High-Dimensional Text Data Visualization

Paweł Jarosz Bartłomiej Walczak Bartosz Hanc

Streszczenie

W projekcie przedstawiono podejście do analizy i wizualizacji danych tekstowych o wysokiej wymiarowości na przykładzie zbioru News Category Dataset. Reprezentacje wektorowe zostały obliczone przy użyciu modelu BERT, a następnie poddane redukcji wymiarowości z wykorzystaniem algorytmów UMAP, t-SNE, PaCMAP i TriMAP. Dokonano również analizy tematycznej (ang. *topic modeling*) z użyciem algorytmu LDA (Latent Dirichlet Allocation), co pozwoliło na identyfikację dominujących wątków w zbiorze oraz analizę ich popularności w czasie. Wyniki zostały zintegrowane w formie interaktywnej aplikacji Streamlit.

1 Wstęp

W dobie rosnącej ilości danych tekstowych, takich jak artykuły prasowe, posty w mediach społecznościowych czy recenzje produktów, coraz większe znaczenie zyskują metody umożliwiające ich efektywną analizę i wizualizację. Surowy tekst, ze względu na swoją nieustrukturyzowaną naturę oraz wysoką wymiarowość po przekształceniu do formy numerycznej, stanowi wyzwanie analityczne. Celem niniejszego projektu było zastosowanie nowoczesnych metod wizualizacji danych tekstowych w celu zrozumienia struktury i tematyki dużego zbioru dokumentów.

W projekcie wykorzystano News Category Dataset – zbiór artykułów prasowych podzielonych na różne kategorie tematyczne. Reprezentacje wektorowe tekstów zostały obliczone, korzystając z modelu językowego BERT (Bidirectional Encoder Representations from Transformers), umożliwiającego uchwycenie głębszego kontekstu treści. Tak przygotowane dane wektorowe zostały następnie poddane redukcji wymiarowości z wykorzystaniem algorytmów takich jak UMAP, t-SNE, PaCMAP oraz TriMAP, co pozwoliło na ich wizualną analizę w przestrzeni dwu- i trójwymiarowej.

Integralnym elementem projektu było zastosowanie analizy tematycznej (ang. *topic modeling*), które pozwala na automatyczne wykrycie ukrytych struktur tematycznych w zbiorze tekstów. W tym celu wykorzystano klasyczny algorytm LDA (Latent Dirichlet Allocation), który przypisuje dokumentom rozkłady prawdopodobieństwa przynależności do różnych tematów. Połączenie wyników modelowania tematów z wizualizacją danych umożliwiło identyfikację powiązań między artykułami, a także śledzenie zmian tematycznych w czasie.

Całość została zintegrowana w interaktywnej aplikacji Streamlit, która umożliwia użytkownikowi dynamiczną eksplorację danych, wizualizacji oraz tematów. Projekt stanowi

praktyczny przykład wykorzystania metod przetwarzania języka naturalnego (NLP), modelowania tematów oraz wizualizacji w analizie dużych korpusów tekstowych, umożliwiając zarówno eksplorację danych, jak i interpretację wyników w sposób intuicyjny i przystępny.

1.1 Użyty zbiór danych

Do realizacji projektu wykorzystano zbiór danych News Category Dataset, który jest dostępny na platformie Kaggle[1]. Zbiór ten zawiera ponad 200 tysięcy nagłówków wiadomości z amerykańskiego portalu informacyjnego HuffPost, zebranych w okresie od 2012 do 2022 roku. Każdy wpis w zbiorze danych stanowi pojedynczy artykuł, reprezentowany głównie przez jego nagłówek, kategorię tematyczną, datę publikacji oraz krótki opis. Struktura danych obejmuje następujące kluczowe kolumny:

- `category` – przypisana kategoria tematyczna (np. polityka, rozrywka, technologia, sport itp.),
- `headline` – tekstowy nagłówek artykułu,
- `authors` – autor lub autorzy artykułu,
- `link` – oryginalny URL do artykułu,
- `short_description` – krótki opis lub wstęp artykułu,
- `date` – data publikacji.

Łącznie zbiór zawiera około 210 000 rekordów, podzielonych na 41 różnych kategorii, co czyni go idealnym materiałem do testowania metod przetwarzania języka naturalnego (NLP), klasyfikacji tekstu, analizy tematów oraz wizualizacji danych tekstowych. W projekcie skoncentrowano się głównie na kolumnach `headline`, `short_description`, `category` i `date`, które zostały użyte do osadzenia tekstu, identyfikacji tematów oraz analizy ich popularności w czasie. Poniżej zamieszczono przykładowy rekord z wykorzystanego zbioru danych.


```
"root":{
    "link": "https://www.huffpost.com/entry/covid-boosters-uptake-
   us_n_632d719ee4b087fae6feaac9"
    "headline": "Over 4 Million Americans Roll Up Sleeves For
   Omicron-Targeted COVID Boosters"
    "category": "U.S. NEWS"
    "short_description": "Health experts said it is too early to
   predict whether demand would match up with the 171 million doses
    of the new boosters the U.S. ordered for the fall."
    "authors": "Carla K. Johnson, AP"
    "date": "2022-09-23"
}
```


[1] https://www.kaggle.com/datasets/rmisra/news-category-dataset

2 Implementacja

2.1 Przetwarzanie danych

Artykuły z pliku `JSON Lines` wczytywane są strumieniowo, a datę `YYYY-MM-DD` zamieniamy na słownik {`year`,`month`,`day`}, co później ułatwia grupowanie czasowe.

```
with open(path) as f:
    articles = [json.loads(l) for l in f]
for a in articles:
    y, m, d = map(int, a["date"].split("-"))
    a["date"] = {"year": y, "month": m, "day": d}
```

Tokenizacja, usunięcie stop-słów i lematyzacja (NLTK) redukują szum i zmniejszają wielkość korpusu. Wynik trafia do pól `processed_headline` i `processed_short_description`. Po złączeniu obu pól każdy artykuł kodowany jest 768-wymiarowym wektorem przy użyciu modelu `distilbert-base-nli-mean-tokens`. Macierz osadzeń zapisujemy do pliku `.npy`, natomiast wzbogacone rekordy do `_processed.json`:

```
texts = [" ".join(a["processed_headline"] +
                  a["processed_short_description"])
         for a in articles]
emb = SentenceTransformer("distilbert-base-nli-mean-tokens").encode
    (
         texts, show_progress_bar=True)
np.save("..._bert_embeddings.npy", emb)
```

Tak przygotowane dane można bezpośrednio poddać redukcji wymiarów (UMAP/t-SNE) i wizualizacji.

2.2 Charakterystyka tematów LDA

Latent Dirichlet Allocation (LDA) modeluje każdy dokument jako mieszaninę ukrytych tematów, a każdy temat jako rozkład prawdopodobieństwa nad słowami. Po wytrenowaniu otrzymujemy dwie macierze:

- $\mathbf{\Theta}_{D\times K}$ – udział tematów $k = 1, \ldots, K$ w poszczególnych dokumentach $d = 1, \ldots, D$,
- $\mathbf{\Phi}_{K\times V}$ – rozkłady słów z słownika (rozmiar V) w każdym temacie.

Temat odczytujemy, wypisując n słów o najwyższych wartościach ϕ_kv. Przykładowo, jeśli w temacie dominują tokeny *election*, *poll*, *vote*, interpretujemy go jako „polityka wyborcza”. LDA zakłada, że kolejność słów jest pomijalna (model *bag-of-words*) oraz że rozkład tematów w dokumencie i słów w temacie podlega rozkładom Dirichleta. W naszym kodzie korzystamy z LDA dostarczonego przez paczkę *gensin*.

```
from gensim.corpora import Dictionary
from gensim.models.ldamodel import LdaModel

...

def lda_topics(tokenized_texts):
```

```
    dictionary = Dictionary(tokenized_texts)
    corpus = [dictionary.doc2bow(text) for text in tokenized_texts]
    lda = LdaModel(
        corpus=corpus,
        num_topics=num_topics,
        id2word=dictionary,
        passes=8,
        random_state=42
    )
```

2.3 Redukcja wymiarowości

W kodzie zawarliśmy funkcje `reduce_dimensions`, która odpowiedzialna jest za zredukowanie wymiarowości wejściowych danych. Wykorzystujemy ją, aby zwizualizować dane w dwóch, lub trzech wymiarach. Skorzystaliśmy z czterech metod:

- UMAP
- TSNE
- PacMAP
- TriMap

Dla każdej z użytych metod wizualizacji daliśmy możliwość wyboru parametrów wizualizacji przy użyciu panelu bocznego. Tabela 1 przedstawwia dostępne parametry, zakresy ich możliwych wartości oraz wartości domyślne, wykorzystywane dla każdej z czterech obsługiwanych technik. Przygotowaliśmy wrapper, który jako argumenty przyjmuje nazwę metody oraz paramtry i zwraca dane zredukowane do zadanej liczby wymiarów.

```
def reduce_dimensions(X, method, params, n_components):
    match method.lower():
        case "umap":
            reducer = UMAP(n_components=n_components, random_state
   =42, **params)
        case "tsne":
            reducer = TSNE(n_components=n_components, random_state
   =42, **params)
        case "pacmap":
            reducer = pacmap.PaCMAP(n_components=n_components, **
   params)
        case "trimap":
            reducer = trimap.TRIMAP(n_dims=n_components, **params)
        case _:
            raise ValueError(f"Unknown reduction method: {method}")
    return reducer.fit_transform(X)
```

Wyniki redukcji są zapisywane i użyte podczas generacji wykresów.

2.4 Wizualizacja

Kilka słow o streamlit

Tabela 1: Dostępne parametry dla poszczególnych metod redukcji wymiarowości

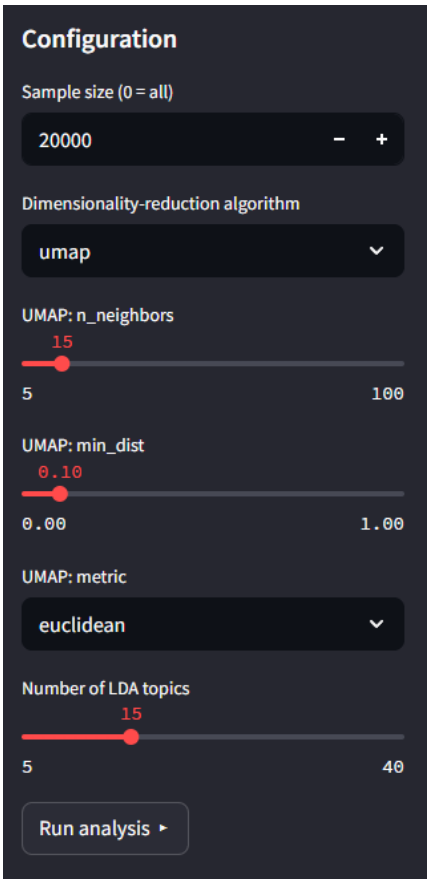
UMAP		
Parametr	Zakres / Wartości	Domyślna
`n_neighbors`	5–100 (slider)	15
`min_dist`	0.0–1.0 (slider)	0.1
`metric`	{euclidean, manhattan, cosine}	euclidean
t-SNE		
Parametr	Zakres / Wartości	Domyślna
`perplexity`	5–50 (slider)	30
`learning_rate`	10–1000 (slider)	200
`metric`	{euclidean, manhattan, cosine}	euclidean
PaCMAP		
Parametr	Zakres / Wartości	Domyślna
`n_neighbors`	5–100 (slider)	10
`MN_ratio`	0.0–1.0 (slider)	0.5
`FP_ratio`	1.0–10.0 (slider)	2.0
`distance`	{euclidean, manhattan, angular}	euclidean
TriMAP		
Parametr	Zakres / Wartości	Domyślna
`n_inliers`	5–100 (slider)	10
`n_outliers`	1–20 (slider)	5
`n_random`	1–10 (slider)	3
`distance`	{euclidean, manhattan, cosine}	euclidean

3 Działanie aplikacji

Aplikacja została zaprojektowana z myślą o interaktywnej obsłudze użytkownika - umożliwia wybór metody redukcji wymiarów, dostosowanie jej parametrów oraz wizualizację wyników w przystępnej formie graficznej. Dzięki wykorzystaniu biblioteki Streamlit, interfejs aplikacji jest przejrzysty i intuicyjny, co ułatwia analizę i porównanie różnych podejść. Poniżej zamieszczono zrzuty ekranu obrazujące elementy interfejsu użytkownika.

Panel konfiguracyjny

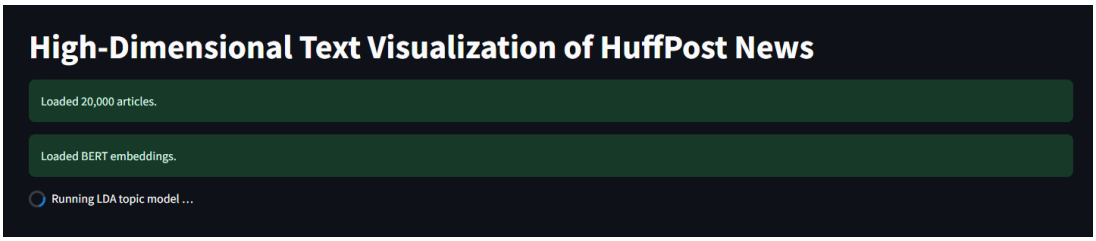
Lewą część okna aplikacji zajmuje panel umożliwiający konfigurację. Pozwala on wybrać liczbę analizowanych artykułów, metodę redukcji wymiarowości wraz z jej parametrami, a także liczbę tematów do analizy tematycznej z użyciem LDA.



Rysunek 1: Panel konfiguracyjny aplikacji

Przebieg analizy

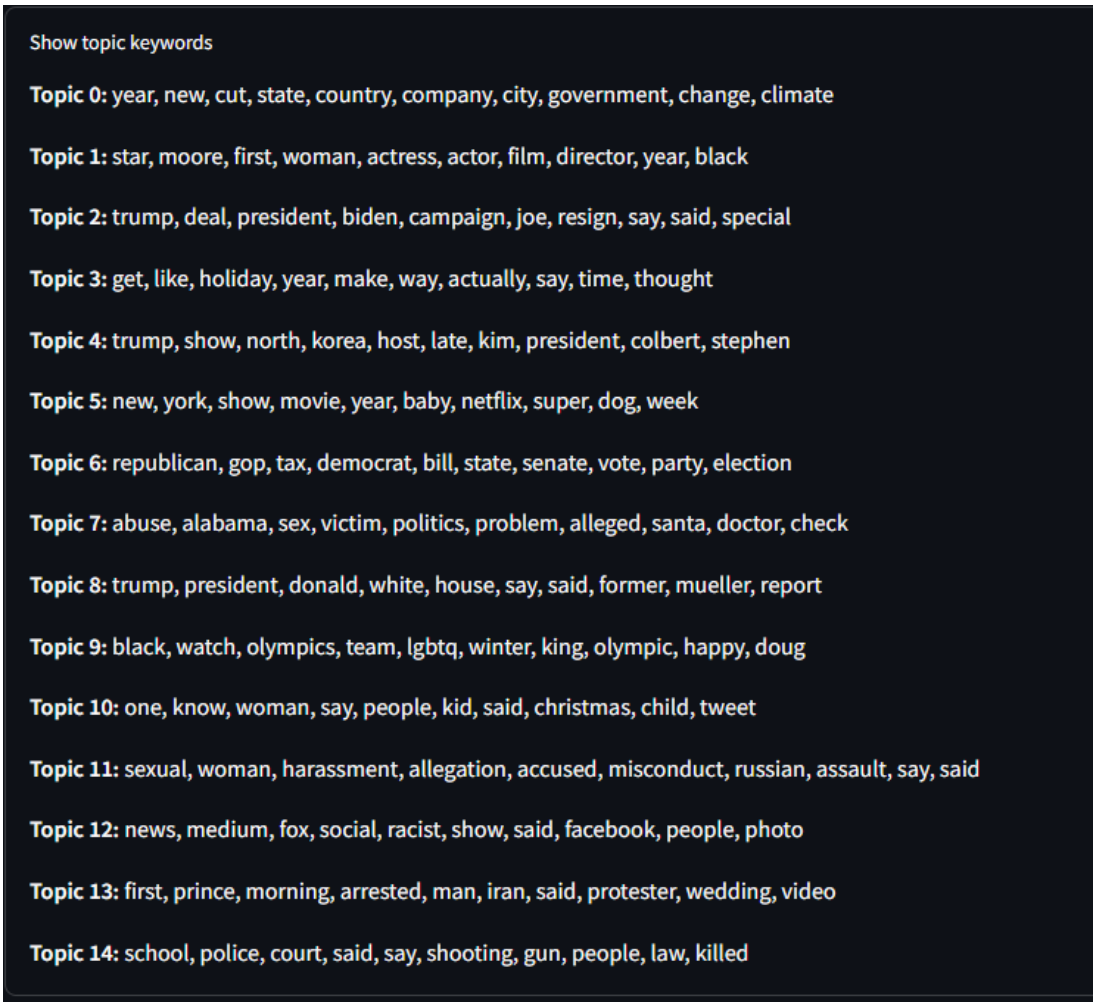
Po uruchomieniu analizy, w głównej części aplikacji użytkownik jest informowany o przebiegu kolejnych etapów.



Rysunek 2: Informacje o przebiegu analizy w głównej części aplikacji

Analiza tematów

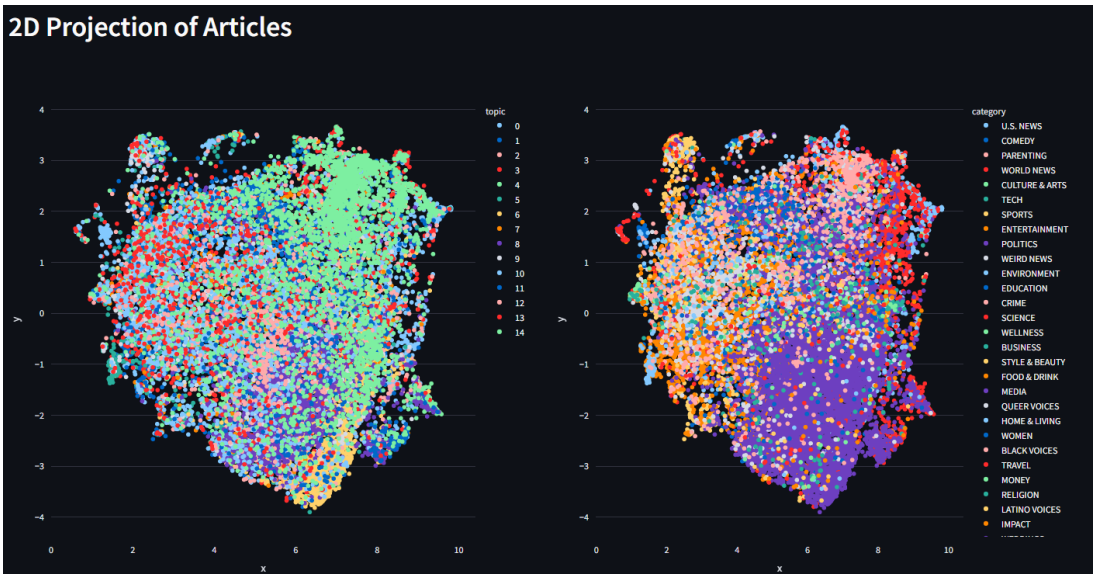
Po wykonaniu metody LDA możemy zobaczyć jej wyniki w formie listy. Prezentowane są słowa kluczowe przypisane do poszczególnych tematów.



Rysunek 3: Wynik analizy tematów

Projekcja 2D

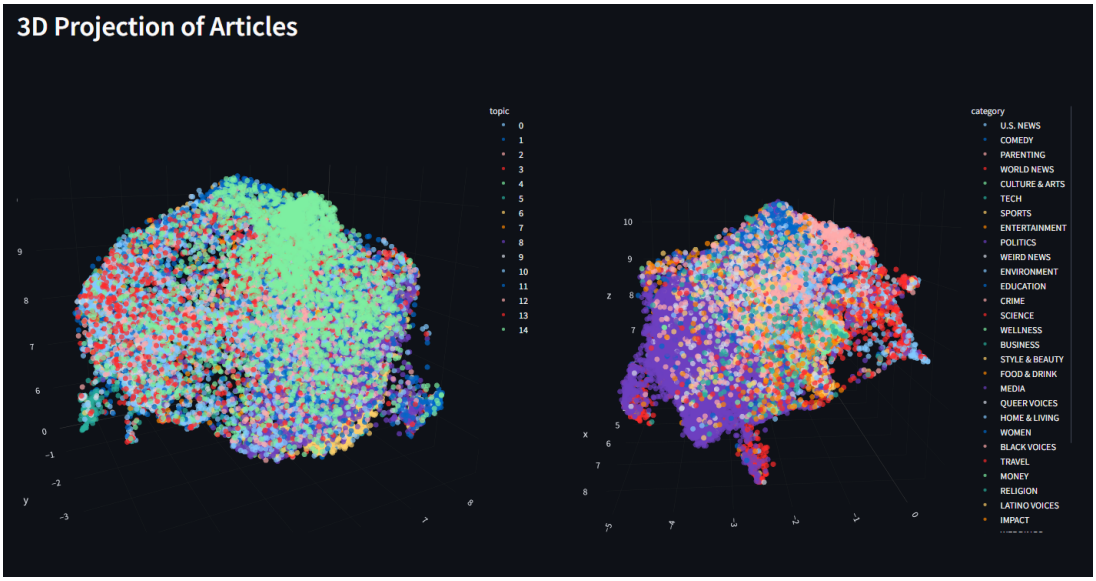
Niżej w aplikacji prezentowana jest projekcja artykułów z użyciem wybranej metody i parametrów na przestrzeń 2D. Prezentowane są tematy przypisane przez LDA oraz kategorie pochodzące ze zbioru danych. Wykresy umożliwiają przybliżenie wybranego fragmentu poprzez zaznaczenie go myszką.



Rysunek 4: Projekcja 2D artykułów

Projekcja 3D

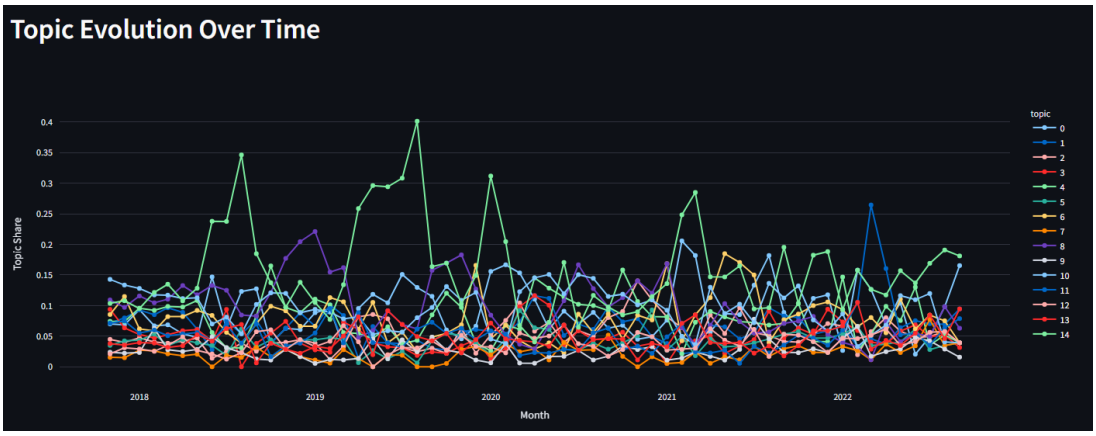
Dostępna jest również analogiczna projekcja na przestrzeń trójwymiarową. Wykres 3D umożliwia obracanie oraz przybliżanie w celu jego efektywnej eksploracji.



Rysunek 5: Projekcja 3D artykułów

Ewolucja tematów w czasie

Aplikacja prezentuje również wykres przedstawiający zmiany popularności tematów artykułów w czasie. Pozwala to np. na wykrycie tematów osiągających wysoką popularność chwilowo lub cyklicznie.



Rysunek 6: Wykres ewolucji tematów w czasie

4 Podsumowanie

Zrealizowany projekt stanowi kompleksowe podejście do eksploracyjnej analizy dużego zbioru dokumentów tekstowych. Łącząc nowoczesne metody reprezentacji semantycznej, takie jak osadzenia BERT, z technikami redukcji wymiarowości i modelowaniem tematów, udało się stworzyć narzędzie umożliwiające intuicyjne zrozumienie struktury i tematyki danych tekstowych.

Wizualizacje uzyskane dzięki algorytmom takim jak UMAP, t-SNE, PaCMAP i TriMap pozwoliły na uchwycenie relacji między dokumentami w niskowymiarowej przestrzeni, ułatwiając identyfikację skupisk tematycznych oraz anomalii. Zastosowanie modelu LDA umożliwiło natomiast przybliżenie ukrytej struktury tematycznej zbioru danych.

Stworzona aplikacja Streamlit integruje wszystkie etapy analizy w jednym środowisku, oferując użytkownikowi możliwość wyboru techniki, dostosowywania parametrów i interpretacji wyników bez konieczności bezpośredniego ingerowania w kod.

Projekt pokazuje, że połączenie metod NLP, modelowania tematów i wizualizacji nie tylko zwiększa przejrzystość i użyteczność analizy tekstu, ale może także przyczynić się do lepszego zrozumienia i wykorzystania dużych, nieustrukturyzowanych zbiorów danych w praktyce.