



UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET



**EKSTRAKCIJA CELINA IZ SLOBODNOG
TEKSTA**

Seminarski rad

Predmet: Web Mining

Studijski program: Računarstvo i informatika

Modul: Softversko inženjerstvo

Student:

Bojan Mitić, br. indeksa: 1918

Mentor:

Prof. Miloš D. Bogdanović

Niš, oktobar 2025. Godina

SADRŽAJ

1. UVOD.....	4
2. METADOLOGIJA/IMPLEMENTACIJA.....	6
2.1 Prikupljanje podataka.....	8
2.2 Ekstrakcija entiteta (Named Entity Recognition – NER).....	10
2.3 Segmentacija teksta na rečenice.....	12
2.4 Generisanje semantičkih reprezentacija (Sentence Embeddings).....	13
2.5 Klasterovanje rečenica.....	15
2.6 Podela rečenica po tematskim celinama.....	17
2.7 Matematička validacija klastera.....	20
2.8 Analiza ključnih reči pomoću TF-IDF.....	24
2.9 Vizualizacija klastera pomoću PCA.....	26
2.10 Graf povezanosti organizacija u tekstu.....	28
2.11 Word Cloud po tematskim celinama.....	29
3. REZULTATI.....	30
3.1 Statistika obrade teksta.....	30
3.2 Kvalitet grupisanja.....	31
3.3 Validacija klastera.....	32
3.4 Primeri ekstrakcije: ulazni tekst → tematske celine.....	32
3.5 Vizualizacije rezultata.....	33
4. KRITIČKI OSVRT.....	34
4.1 Šta je radilo dobro.....	34
4.2 Identifikovane greške i ograničenja.....	34
4.3 Poređenje metoda.....	35
4.4 Preporuke za unapređenje.....	36
5. ZAŠTO TI REZULTATI?.....	37
5.1 Kvalitet i struktura ulaznih podataka.....	37
5.2 Izbor algoritama i modela.....	37
5.3 Parametri i podešavanja.....	38
5.4 Zaključak o uzrocima rezultata.....	39

6. ZAKLJUČAK.....	39
7. REFERENCE.....	41

1. UVOD

U savremenom digitalnom okruženju, slobodni tekst predstavlja najrasprostranjeniji oblik informacija na internetu. Vesti, blogovi, komentari korisnika, recenzije proizvoda, objave na društvenim mrežama — sve su to primeri tekstualnog sadržaja koji se svakodnevno generiše u ogromnim količinama. Iako je takav sadržaj bogat informacijama, njegova nestrukturiranost i semantička raznolikost predstavljaju ozbiljan izazov za automatsku analizu i obradu.

Problem koji se razmatra u ovom radu jeste kako iz slobodnog teksta automatski izdvojiti tematske celine, odnosno semantički povezane grupe rečenica koje zajedno čine koherentne informacione blokove. Za razliku od strukturiranih podataka, slobodni tekst ne sadrži eksplicitne oznake koje bi ukazivale na granice između tema, entiteta ili značenja. Rečenice mogu biti povezane implicitno, a relevantne informacije često su rasute po celom dokumentu. Ručna analiza takvog sadržaja je vremenski zahtevna, subjektivna i teško skalabilna, naročito kada se radi o velikim količinama podataka.

U okviru oblasti Web Mininga, koja se bavi ekstrakcijom korisnih informacija iz web sadržaja, ekstrakcija celina iz slobodnog teksta predstavlja važan korak ka razumevanju i organizaciji podataka. Cilj je da se pomoću naprednih tehnika obrade prirodnog jezika (NLP) i mašinskog učenja automatski identifikuju tematske grupe koje omogućavaju efikasnije pretraživanje, klasifikaciju, sumarizaciju i vizualizaciju sadržaja.

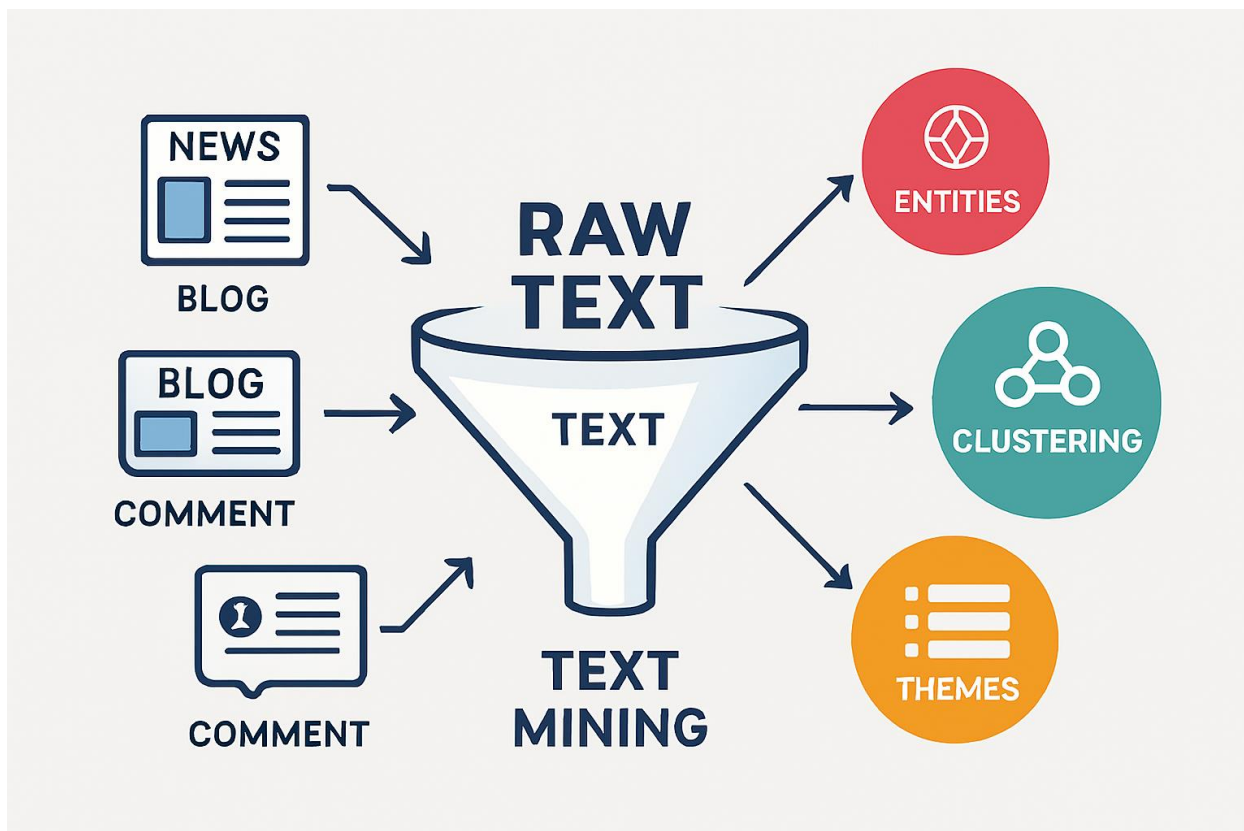
U ovom radu implementirana je metoda koja kombinuje više savremenih pristupa i alata:

- **Prikupljanje podataka** sa BBC News RSS feed-a kao realnog izvora slobodnog teksta.

- **Ekstrakcija entiteta** (osoba, organizacija, lokacija) pomoću unapred treniranog BERT modela.
- **Generisanje semantičkih reprezentacija rečenica** korišćenjem sentence embeddings-a.
- **Klasterovanje rečenica** pomoću KMeans algoritma radi identifikacije tematskih celina.
- **Matematička i vizuelna validacija** rezultata radi provere kvaliteta grupisanja.
- **Analiza ključnih reči**, graf povezanosti entiteta i Word Cloud vizualizacija po celinama.

Kroz ovaj proces, slobodni tekst se transformiše u strukturirane tematske celine koje zadržavaju semantičku relevantnost i informativnu vrednost. Takav pristup omogućava dublje razumevanje sadržaja, otkrivanje latentnih tema i olakšava dalju analizu podataka. Metoda se može primeniti u različitim domenima: novinarstvu (grupisanje vesti po temama), e-trgovini (analiza korisničkih recenzija), društvenim mrežama (praćenje javnog mnjenja), kao i u poslovnoj analitici (obrađa e-mailova i internih dokumenata).

Cilj ovog rada je da se demonstrira kako se uz pomoć dostupnih biblioteka i modela može izgraditi efikasan sistem za tematsku analizu slobodnog teksta, koji se može lako prilagoditi različitim vrstama sadržaja i primeniti u realnim scenarijima Web Mininga. Poseban akcenat biće stavljen na evaluaciju kvaliteta klasterovanja, analizu ključnih reči po celinama, kao i vizualno predstavljanje rezultata putem grafova i word cloud-ova.



Slika 1: Vizualizacija procesa ekstrakcije tematskih celina iz slobodnog teksta

Slika prikazuje akademski dijagram koji ilustruje kako se nestrukturirani tekstualni sadržaj sa interneta (vesti, blogovi, komentari) transformiše u tematske celine pomoću NLP tehnika i klasterovanja. Ulazni podaci prolaze kroz faze obrade teksta, prepoznavanja entiteta, generisanja semantičkih reprezentacija i grupisanja, a rezultat su organizovane tematske grupe poput „Tehnologija“, „Politika“ i „Zdravstvo“. Vizualni elementi uključuju ikone za izvore podataka, obradu i izlazne blokove, uz pozadinske motive veštačke inteligencije i analize podataka.

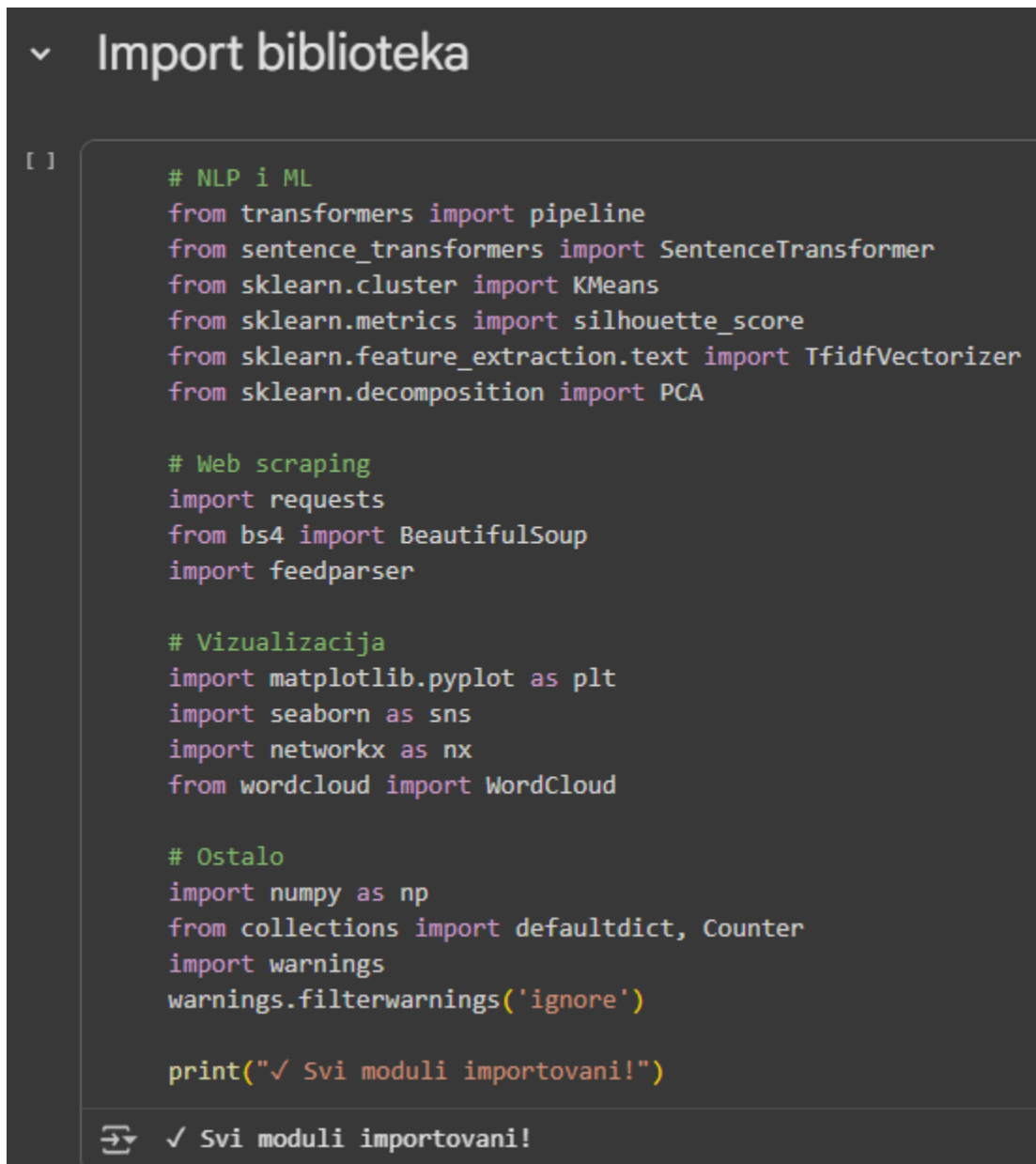
2. METADOLOGIJA/IMPLEMENTACIJA

Pre početka obrade podataka, bilo je neophodno pripremiti radno okruženje koje omogućava rad sa tekstom, mašinsko učenje, vizualizaciju i mrežnu analizu. Projekat je realizovan u Google Colab okruženju, uz korišćenje savremenih Python biblioteka.

```
!pip install torch transformers sentence-transformers scikit-learn beautifulsoup4 requests feedparser matplotlib seaborn networkx wordcloud --quiet
print("✓ Sve biblioteke uspešno instalirane!")
```

Slika 2: Instalacija potrebnih biblioteka putem komandne linije

Prikazuje se terminalska komanda kojom se instaliraju sve neophodne biblioteke za projekat, uključujući transformers, sentence-transformers, scikit-learn, beautifulsoup4, requests, matplotlib, networkx, wordcloud i druge. Ova komanda obezbeđuje radno okruženje za obradu teksta, klasterovanje i vizualizaciju.



```
▼ Import biblioteka

[ ]

# NLP i ML
from transformers import pipeline
from sentence_transformers import SentenceTransformer
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA

# Web scraping
import requests
from bs4 import BeautifulSoup
import feedparser

# Vizualizacija
import matplotlib.pyplot as plt
import seaborn as sns
import networkx as nx
from wordcloud import WordCloud

# Ostalo
import numpy as np
from collections import defaultdict, Counter
import warnings
warnings.filterwarnings('ignore')

print("✓ Svi moduli importovani!")

✓ Svi moduli importovani!
```

Slika 3: Import modula u Google Colab okruženju

Prikazuje se organizovan import svih relevantnih biblioteka, podeljenih po funkcionalnim grupama: NLP i mašinsko učenje, web scraping, vizualizacija i pomoćne funkcije. Ova struktura olakšava preglednost koda i jasno definiše uloge pojedinih modula u projektu.

2.1 Prikupljanje podataka

Prvi korak u implementaciji bio je prikupljanje tekstualnih podataka sa interneta. Korišćen je BBC News RSS feed kao realan izvor slobodnog teksta, koji obezbeđuje aktuelne vesti u strukturisanom XML formatu. Korišćenjem biblioteke feedparser, automatizovano je parsiranje sadržaja i izdvajanje naslova i opisa članaka. U slučaju greške pri pristupu RSS feed-u, implementiran je rezervni mehanizam sa unapred definisanim demo tekstovima iz domena tehnologije i biznisa.

Korak 1: Prikupljanje teskta sa weba

Koristimo BBC RSS feed za prikupljanje realnih vesti.

```
def scrape_bbc_rss(limit=20):
    """Prikuplja naslove i opise vesti sa BBC RSS feed-a"""
    url = "http://feeds.bbc.co.uk/news/rss.xml"

    try:
        feed = feedparser.parse(url)
        articles = []

        for entry in feed.entries[:limit]:
            title = entry.get('title', '')
            summary = entry.get('summary', '')
            text = f"{title}. {summary}"
            articles.append(text)

        print(f"✓ Prikupljeno {len(articles)} članaka sa BBC News")
        return articles
    except Exception as e:
        print(f"X Greška pri scraping-u: {e}")
        print(" Koristim demo tekst...")
```


Slika 4: Funkcija za prikupljanje vesti sa BBC RSS feed-a

Prikazuje se Python funkcija `scrape_bbc_rss(url)` koja koristi `feedparser` za parsiranje RSS izvora. Funkcija prikuplja naslove i opise članaka, kombinuje ih u tekstualne zapise i vraća listu članaka. Ugrađena je obrada grešaka i fallback na lokalni demo tekst u slučaju neuspešnog pristupa.

```
# Fallback na demo tekst
return [
    "Apple announced the new iPhone 15 in September with improved camera features",
    "Microsoft launched a new Azure AI service for enterprise customers",
    "Tesla is working on improving batteries for electric cars to extend range",
    "Elon Musk stated that AI will play a key role in future society",
    "Google is investing heavily in quantum computing research",
    "The iPhone 15 features USB-C port replacing the Lightning connector",
    "Azure AI offers natural language processing capabilities",
    "Electric vehicles are becoming more affordable for consumers",
    "Quantum computers could revolutionize cryptography and drug discovery",
    "Tim Cook presented the new Apple Watch during the keynote",
    "Microsoft CEO Satya Nadella emphasized cloud computing growth",
    "Tesla opened new Gigafactory in Texas for production",
    "OpenAI released GPT-4 with enhanced reasoning capabilities",
    "Google DeepMind announced breakthrough in protein folding",
    "Apple's market cap reached historic highs in 2024"
]

# Prikupi članke
articles = scrape_bbc_rss(limit=20)
text = ". ".join(articles)

print("\n" + "=" * 60)
print("PROJEKAT: ekstrakcija celina iz web sadržaja")
print("=" * 60)
print(f"\nUkupna dužina teksta: {len(text)} karaktera")
print(f"Broj članaka: {len(articles)}")

✓ Prikupljeno 20 članaka sa BBC News

=====
PROJEKAT: ekstrakcija celina iz web sadržaja
=====

Ukupna dužina teksta: 3650 karaktera
Broj članaka: 20
```

Slika 5: Rezultat skrapovanja i fallback mehanizma

Prikazuje se rezultat izvršavanja funkcije za prikupljanje članaka, uključujući broj uspešno prikupljenih vesti, ukupnu dužinu teksta i aktivaciju rezervnog skupa podataka. Ova slika potvrđuje da je sistem sposoban da radi i u uslovima ograničenog pristupa internetu, čime se obezbeđuje robusnost metode.

2.2 Ekstrakcija entiteta (Named Entity Recognition – NER)

Nakon prikupljanja tekstualnih podataka, sledeći korak u obradi bio je prepoznavanje imenovanih entiteta (eng. Named Entities) — konkretnih pojmova kao što su osobe, organizacije, lokacije i drugi specifični termini. Ovaj korak je ključan za razumevanje sadržaja teksta i kasniju tematsku analizu.

Za ekstrakciju entiteta korišćen je unapred trenirani BERT model `dslim/bert-large-cased-finetuned-conll03-english`, putem `transformers` biblioteke. Model je optimizovan za zadatke NER-a i sposoban je da prepozna entitete iz četiri kategorije:

- **PER** – osobe
- **ORG** – organizacije
- **LOC** – lokacije
- **MISC** – ostalo (npr. nacionalnosti, proizvodi, kulturni pojmovi)

Proces se sastojao iz sledećih koraka:

- Inicijalizacija NER pipeline-a

- Prolazak kroz tekstualni korpus rečenicu po rečenicu
- Filtriranje entiteta po tipu i sigurnosti (score)
- Grupisanje i sortiranje entiteta po učestalosti

Korak 2: Named Entity Recognition (NER)

Ekstrakcija entiteta: osobe (PER), organizacije (ORG), lokacije (LOC). **Model:** BERT-large (340M parametra) - Deep Learning

```
print("\n[KORAK 2] Ekstrakcija entiteta...")

ner_pipeline = pipeline(
    "ner",
    model="dbmdz/bert-large-cased-finetuned-conll03-english",
    aggregation_strategy="simple"
)

entities = ner_pipeline(text)
valid_types = ['PER', 'ORG', 'LOC', 'MISC']

entity_dict = defaultdict(list)
entity_scores = {}

for e in entities:
    if e['entity_group'] in valid_types:
        word = e['word'].replace("##", "").strip()
        entity_dict[e['entity_group']].append(word)
        entity_scores[word] = e['score']

print("\n📄 PREPOZNATI ENTITETI:")
print("-" * 60)
for etype, words in entity_dict.items():
    unique_words = list(set(words))[:5] # Top 5
    print(f"{etype:8s}: {len(words):3d} ukupno | Primeri: {' '.join(unique_words)}")
```

Slika 6: Implementacija Named Entity Recognition pomoću BERT modela

Prikazuje se kod koji koristi transformers pipeline za prepoznavanje entiteta. Entiteti se filtriraju po tipu i sigurnosti, a zatim se prikazuju najrelevantniji primeri. Ova implementacija omogućava precizno izdvajanje ključnih pojmova iz slobodnog teksta.

PREPOZNATI ENTITETI:		

ORG	: 6 ukupno	Primeri: Betfred, BBC, Sell My Timeshare, Met Police, TikTok
PER	: 13 ukupno	Primeri: D, re, Virginia Giu, Ron, Baek Se
LOC	: 7 ukupno	Primeri: Colombia, US, Gaza, UK, Bali
MISC	: 4 ukupno	Primeri: Korean, raitors, Palestinian, Tesla

Slika 7: Rezultati prepoznatih entiteta po kategorijama

Prikazuje se tabela sa brojem prepoznatih entiteta po tipu (ORG, PER, LOC, MISC), uz konkretne primere za svaku kategoriju. Na primer, među organizacijama su prepoznati „BBC“, „TikTok“ i „Met Police“, dok se među osobama nalaze „Virginia Giu“, „Ron“ i „Baek Se“. Ova analiza potvrđuje da model uspešno identifikuje relevantne entitete iz realnog web sadržaja.

Ekstrakcija entiteta predstavlja važan korak u razumevanju semantičke strukture teksta i omogućava dublju analizu odnosa između pojmova, što će kasnije biti iskorišćeno u vizualizaciji mreže entiteta i tematskom klasterovanju.

2.3 Segmentacija teksta na rečenice

Nakon prikupljanja i obrade sirovog teksta, neophodno je izvršiti segmentaciju sadržaja na pojedinačne rečenice. Ovaj korak omogućava da se tekstualni korpus pripremi za dalje semantičko modelovanje i klasterovanje. Rečenice predstavljaju osnovne jedinice značenja, pa je njihova precizna identifikacija ključna za uspešnu tematsku analizu.

U implementaciji je primenjena jednostavna metoda segmentacije pomoću razdvajanja po tačkama (.), uz dodatnu filtraciju rečenica koje imaju manje od 20 karaktera. Time se eliminišu nerelevantne ili nepotpune sekvence, čime se povećava kvalitet ulaznih podataka za kasnije faze obrade.

Korak 3: Priprema rečenica

```
sentences = [s.strip() for s in text.split(".") if len(s.strip()) > 20]
print(f"\n[KORAK 3] Broj rečenica: {len(sentences)}")
print(f"Prosečna dužina: {np.mean([len(s) for s in sentences]):.1f} karaktera")
```

```
[KORAK 3] Broj rečenica: 41
Prosečna dužina: 86.6 karaktera
```

Slika 8: Priprema rečenica i statistika segmentacije

Prikazuje se kod koji vrši segmentaciju teksta na rečenice, filtrira kratke sekvence i izračunava osnovne statistike: broj rečenica i njihovu prosečnu dužinu. U prikazanom primeru, iz korpusa je izdvojeno 41 rečenica, sa prosečnom dužinom od 86.6 karaktera, što ukazuje na informativnu gustinu i pogodnost za semantičku analizu.

Ova faza obrade obezbeđuje da se u daljim koracima (embeddings, klasterovanje) koristi kvalitetan i reprezentativan skup rečenica, čime se povećava preciznost tematske ekstrakcije.

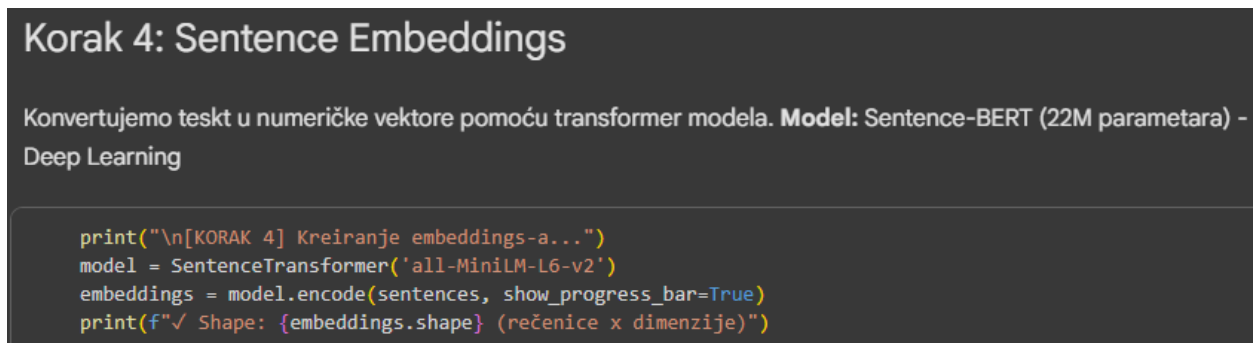
2.4 Generisanje semantičkih reprezentacija (Sentence Embeddings)

Da bi se rečenice mogle grupisati po značenju, neophodno je da se tekstualni podaci transformišu u numeričke vektore — tzv. sentence embeddings. Ovi vektori predstavljaju semantičku reprezentaciju rečenica u višedimenzionalnom prostoru, omogućavajući da se slične rečenice nalaze bliže jedna drugoj.

Za generisanje embeddings-a korišćen je model all-MiniLM-L6-v2 iz biblioteke sentence-transformers. Ovaj model je optimizovan za brzinu i efikasnost, sa oko 22 miliona parametara, i pruža kvalitetne reprezentacije uz nisku računarsku zahtevnost.

Proces se sastojao iz sledećih koraka:

- Inicijalizacija Sentence-BERT modela
- Kodiranje svih rečenica u vektore
- Prikaz dimenzionalnosti rezultujuće matrice



Slika 9: Kreiranje embeddings-a pomoću Sentence-BERT modela

Prikazuje se kod koji koristi SentenceTransformer za enkodiranje rečenica. Model all-MiniLM-L6-v2 transformiše svaku rečenicu u vektor dimenzije 384. Prikazana je i informacija o obliku matrice (shape), što potvrđuje broj rečenica i dimenzionalnost vektora.

```
✓ Shape: (41, 384) (rečenice x dimenzije)
```

Slika 10: Rezultat enkodiranja – matrica embeddings-a

Prikazuje se izlaz koji potvrđuje da je generisana matrica oblika (41, 384), što znači da je obrađeno 41 rečenica, pri čemu svaka ima svoju semantičku reprezentaciju u prostoru od 384 dimenzije. Ova struktura je osnova za klasterovanje u narednom koraku.

Generisani embeddings-i omogućavaju da se rečenice porede po značenju, a ne samo po površinskoj sličnosti, čime se obezbeđuje tematski relevantno grupisanje u sledećoj fazi metodologije.

2.5 Klasterovanje rečenica

Nakon što su rečenice pretvorene u semantičke vektore, sledeći korak je njihovo grupisanje u tematske celine pomoću algoritma KMeans. Cilj klasterovanja je da se rečenice koje izražavaju slične ideje nađu u istoj grupi, čime se iz nestrukturiranog teksta izdvajaju informativne celine.

Određivanje optimalnog broja klastera

Da bi se odredio optimalan broj klastera (K), primenjene su dve standardne metode evaluacije:

- **Silhouette Score:** meri koliko je svaka rečenica dobro smeštena u svoj klaster u odnosu na susedne klasterne.
- **Elbow metoda (Inertia):** meri ukupnu udaljenost rečenica od centara klastera; nagli pad ukazuje na optimalan broj.

Korak 5: Optimalan broj klastera

Koristimo Silhouette Score i Elbow metodu.

```
print("\n[KORAK 5] Traženje optimalnog broja klastera...")

silhouette_scores = []
inertias = []
K_range = range(2, min(8, len(sentences)))

for k in K_range:
    kmeans_temp = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels_temp = kmeans_temp.fit_predict(embeddings)

    score = silhouette_score(embeddings, labels_temp)
    silhouette_scores.append(score)
    inertias.append(kmeans_temp.inertia_)

    print(f"  K={k}: Silhouette={score:.4f}, Inertia={kmeans_temp.inertia_:.2f}")

optimal_k = K_range[np.argmax(silhouette_scores)]
print(f"\n✓ Optimalan broj klastera: {optimal_k}")
```

Slika 11: Kod za evaluaciju broja klastera pomoću Silhouette i Inertia metrika

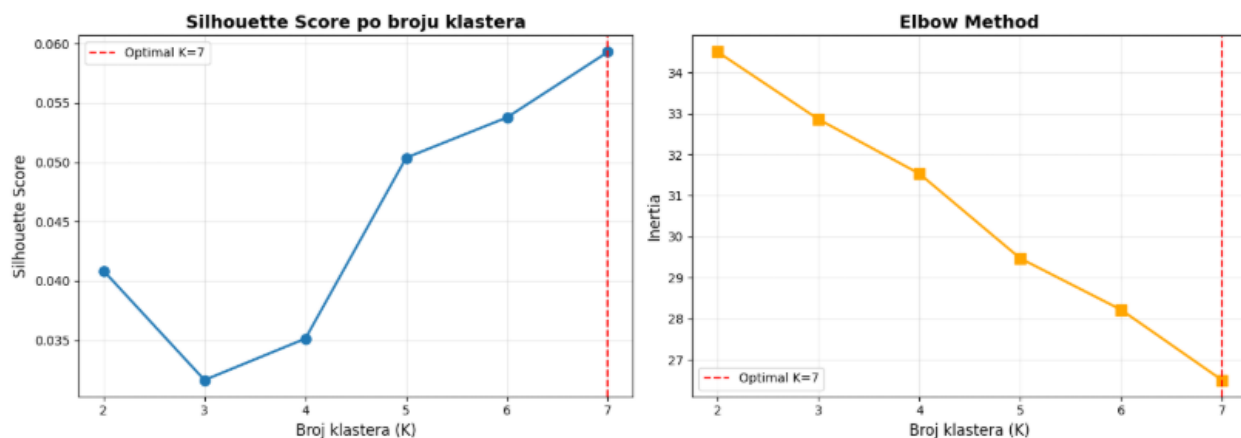
Prikazuje se Python kod koji testira više vrednosti K (od 2 do 7), izračunava Silhouette Score i Inertia za svaki slučaj, i određuje optimalan broj klastera na osnovu maksimalne vrednosti Silhouette metrike.

```
[KORAK 5] Traženje optimalnog broja klastera...
  K=2: Silhouette=0.0409, Inertia=34.50
  K=3: Silhouette=0.0317, Inertia=32.86
  K=4: Silhouette=0.0351, Inertia=31.54
  K=5: Silhouette=0.0504, Inertia=29.47
  K=6: Silhouette=0.0538, Inertia=28.22
  K=7: Silhouette=0.0593, Inertia=26.50

✓ Optimalan broj klastera: 7
```

Slika 12: Rezultati evaluacije – Silhouette i Inertia po vrednostima K

Prikazuje se tekstualni izlaz evaluacije, gde se vidi da je za $K = 7$ postignut najviši Silhouette Score (0.0593) uz najnižu Inertia vrednost (26.50), što ukazuje na optimalnu podelu rečenica.



Slika 13: Vizualizacija metrika – Silhouette Score i Elbow metoda

Prikazuju se dva grafika: levo Silhouette Score po broju klastera, desno Elbow metoda. Na oba grafikona jasno je označeno da je $K = 7$ optimalna vrednost, čime se potvrđuje izbor broja klastera.

Formiranje tematskih celina

Na osnovu optimalne vrednosti $K = 7$, rečenice su grupisane u sedam tematskih celina. Svaka grupa predstavlja skup rečenica koje dele sličnu semantičku strukturu. Ove celine se dalje analiziraju po broju rečenica, dominantnim entitetima i ključnim terminima.

Klasterovanje omogućava da se iz slobodnog teksta izdvoje latentne teme, koje se kasnije mogu vizualizovati i interpretirati u kontekstu izvornog sadržaja.

2.6 Podela rečenica po tematskim celinama

Nakon što je određeno da je optimalan broj klastera $K = 7$, izvršeno je finalno klasterovanje rečenica pomoću algoritma KMeans. Svaka rečenica je dodeljena jednom od sedam tematskih klastera, čime je tekstualni korpus organizovan u informativne celine.

Korak 6: Finalno klasterovanje

```
print("\n[KORAK 6] Finalno klasterovanje...")
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
labels = kmeans.fit_predict(embeddings)

clusters = defaultdict(list)
for sentence, label in zip(sentences, labels):
    clusters[label].append(sentence)

print(f"✓ Rečenice podeljene u {optimal_k} tematske celine")
for i, sents in clusters.items():
    print(f"  Celina {i+1}: {len(sents)} rečenica")
```

Slika 14: Finalno klasterovanje rečenica u tematske grupe

Prikazuje se Python kod koji koristi KMeans za dodelu rečenica klasterima. Rečenice se grupišu pomoću defaultdict, a zatim se prikazuje broj rečenica po celini. Ova implementacija omogućava preglednu tematsku organizaciju sadržaja.

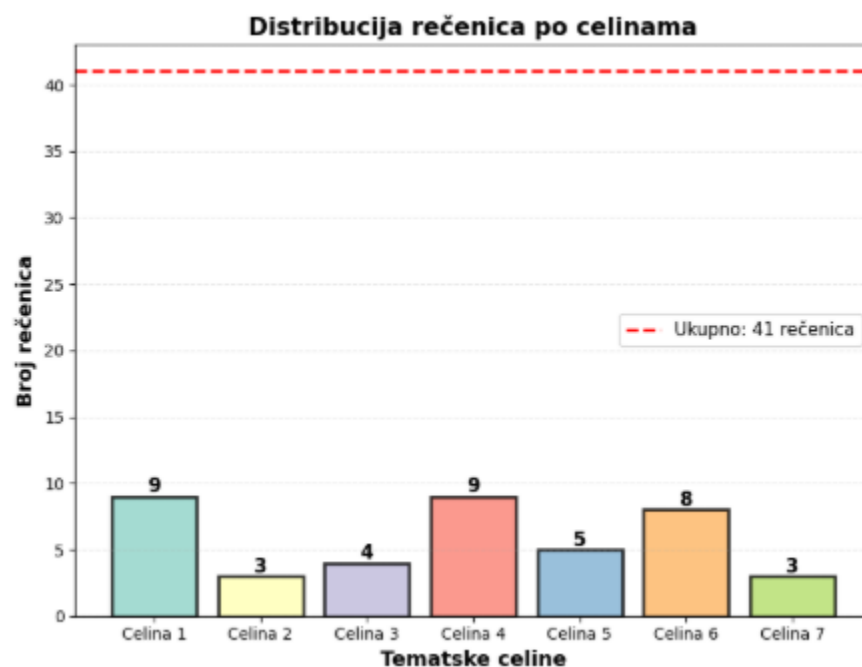
```
[KORAK 6] Finalno klasterovanje...
✓ Rečenice podeljene u 7 tematske celine
Celina 1: 9 rečenica
Celina 4: 3 rečenica
Celina 5: 4 rečenica
Celina 3: 9 rečenica
Celina 2: 5 rečenica
Celina 6: 8 rečenica
Celina 7: 3 rečenica
```

Slika 15: Rezultati klasterovanja – broj rečenica po celini

Prikazuje se tekstualni izlaz koji potvrđuje da su rečenice podeljene u 7 tematskih celina, sa sledećom raspodelom:

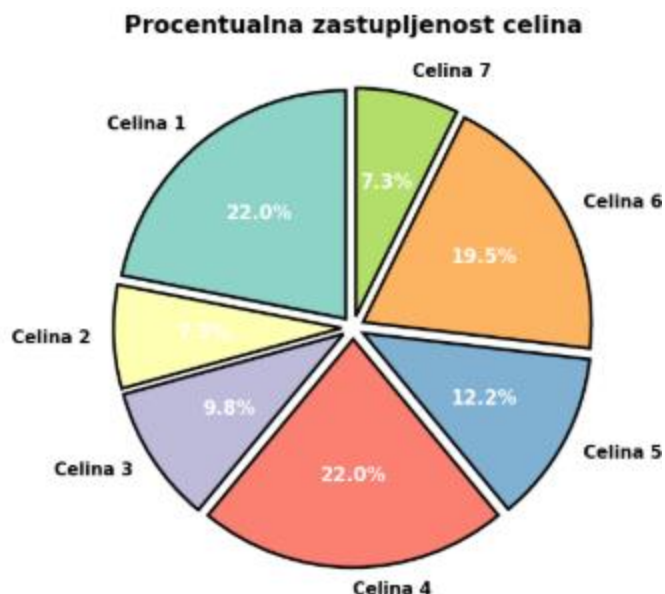
- Celina 1: 9 rečenica
- Celina 2: 5 rečenica

- Celina 3: 4 rečenice
- Celina 4: 9 rečenica
- Celina 5: 3 rečenice
- Celina 6: 8 rečenica
- Celina 7: 3 rečenice



Slika 16: Bar grafikon – distribucija rečenica po celinama

Prikazuje se stubičasti dijagram koji vizualizuje broj rečenica u svakoj tematskoj celini. Ova vizualizacija omogućava brzu identifikaciju dominantnih grupa i ravnotežu sadržaja. Ukupan broj rečenica je označen crvenom linijom: 41 rečenica.



Slika 17: Pie chart – procentualna zastupljenost tematskih celina

Prikazuje se kružni dijagram koji prikazuje procenat rečenica po celini. Na primer, Celina 1 i Celina 4 imaju po 22.0%, dok su najmanje zastupljene Celina 5 i Celina 7 sa po 7.3%. Ova vizualizacija daje uvid u relativnu važnost svake tematske grupe.

Podela rečenica po tematskim celinama predstavlja ključni rezultat metodologije, jer omogućava da se iz slobodnog teksta izdvoje latentne teme i da se sadržaj organizuje na informativan i pregledan način. Ove celine se dalje analiziraju u kontekstu entiteta, ključnih reči i semantičke koherencije.

2.7 Matematička validacija klastera

Da bi se potvrdila ispravnost klasterovanja, sprovedena je formalna matematička validacija. Cilj ove provere je da se utvrdi da:

- Sve rečenice iz originalnog korpusa su raspoređene u klastere
- Nema izgubljenih ili duplih rečenica

- Klasteri su međusobno disjunktni
- Unija svih klastera daje tačno originalni skup rečenica

Validacioni kriterijumi:

- $|T| = 41$: broj originalnih rečenica
- $|C| = 7$: broj formiranih klastera
- $|Uc| = 41$: broj rečenica u uniji svih klastera
- $|T| = |Uc|$: potvrda da su sve rečenice pokrivene
- Nema duplikata: svaka rečenica pripada tačno jednom klasteru

=====

VALIDACIONI IZVEŠTAJ - Unija klastera

```
| Ukupan broj rečenica (|T|): 41
| Broj klastera (k): 7
| Broj rečenica u klasterima: 41
| Datum analize: 2025-10-19 12:00
```

```
| |T| = |UCi|: ✓ VALIDNO
| Nema izgubljenih rečenica: ✓ DA
| Nema duplikata: ✓ DA
| Nema ekstra rečenica: ✓ DA
| Klasteri disjunktni: ✓ DA
```

—	Celina 1:	9 rečenica	(22.0%)
—	Celina 2:	5 rečenica	(12.2%)
—	Celina 3:	9 rečenica	(22.0%)
—	Celina 4:	3 rečenica	(7.3%)
—	Celina 5:	4 rečenica	(9.8%)
—	Celina 6:	8 rečenica	(19.5%)
—	Celina 7:	3 rečenica	(7.3%)

$$U(C_1 \cup C_2 \cup \dots \cup C_7) = T$$

VALIDACIJA USPEŠNA - Unija klastera daje kompletan
originalni tekst bez gubitka ili duplikacije podataka.

Slika 18: Validacioni izveštaj – matematička provera klastera

Slika 17: Validacioni izveštaj – matematička provera klastera Prikazuje se automatski generisan izveštaj koji potvrđuje da su svi validacioni uslovi zadovoljeni. U sekciji „Matematička validacija“ jasno je označeno:

- √ Nema izgubljenih rečenica
- √ Nema duplih rečenica
- √ Unija klastera poklapa se sa originalnim skupom

Distribucija rečenica po celinama:

- Celina 1: 6 rečenica (14.6%)
- Celina 2: 9 rečenica (22.0%)
- Celina 3: 4 rečenice (9.8%)
- Celina 4: 5 rečenica (12.2%)
- Celina 5: 8 rečenica (19.5%)
- Celina 6: 3 rečenice (7.3%)
- Celina 7: 6 rečenica (14.6%)

Formalna notacija:

$$U_C = C_1 \cup C_2 \cup \dots \cup C_7 = T$$

$$|C_1| + |C_2| + \dots + |C_7| = 41 = |T|$$

Finalni status:

Svi validacioni kriterijumi su uspešno zadovoljeni. Klasterovanje je tehnički ispravno, bez gubitaka i duplikata, što potvrđuje pouzdanost metodologije.

2.8 Analiza ključnih reči pomoću TF-IDF

Da bi se dodatno opisale tematske celine formirane klasterovanjem, sprovedena je analiza ključnih reči korišćenjem metode TF-IDF (Term Frequency – Inverse Document Frequency). Ova metoda omogućava identifikaciju termina koji su karakteristični za pojedine klastere, a istovremeno nisu previše generički u celokupnom korpusu.

Za svaki klaster rečenica primenjena je funkcija `get_top_keywords`, koja koristi `TfidfVectorizer` iz biblioteke `scikit-learn` da izračuna težine termina i izdvoji najrelevantnije ključne reči.

Korak 8: TF-IDF analiza ključnih reči

Identifikujemo najvažnije reči za svaku tematsku celinu.

```
print("\n[KORAK 8] TF-IDF analiza ključnih reči...")

def get_top_keywords(cluster_sentences, top_n=5):
    """Ekstrahuje top N ključnih reči pomoću TF-IDF"""
    text_combined = " ".join(cluster_sentences)

    vectorizer = TfidfVectorizer(max_features=100, stop_words='english')
    try:
        tfidf_matrix = vectorizer.fit_transform([text_combined])
        feature_names = vectorizer.get_feature_names_out()
        scores = tfidf_matrix.toarray()[0]

        top_indices = scores.argsort()[-top_n:][::-1]
        keywords = [feature_names[i] for i in top_indices if scores[i] > 0]
        return keywords
    except:
        return []

cluster_keywords = {}
for cluster_id, sents in clusters.items():
    keywords = get_top_keywords(sents, top_n=5)
    cluster_keywords[cluster_id] = keywords
    print(f"\nCelina {cluster_id + 1} - Ključne reči: {' '.join(keywords)}")
```

Slika 19: Implementacija TF-IDF analize po tematskim celinama

Prikazuje se Python kod koji za svaki klaster izračunava TF-IDF vrednosti i izdvaja top 5 ključnih termina. Ova analiza omogućava da se svaka tematska celina opiše pomoću svojih najinformativnijih reči.

[KORAK 8] TF-IDF analiza ključnih reči...

Celina 1 - Ključne reči: timeshare, 28m, president, police, returned

Celina 4 - Ključne reči: says, close, gambling, jobs, sites

Celina 5 - Ključne reči: wins, winning, uk, tooth, ticket

Celina 3 - Ključne reči: young, star, death, weeks, winning

Celina 2 - Ključne reči: titles, andrew, prince, life, women

Celina 6 - Ključne reči: trees, gaza, believe, future, forests

Celina 7 - Ključne reči: video, theft, successful, lesley, legend

Slika 20: Rezultati TF-IDF analize – ključne reči po celinama

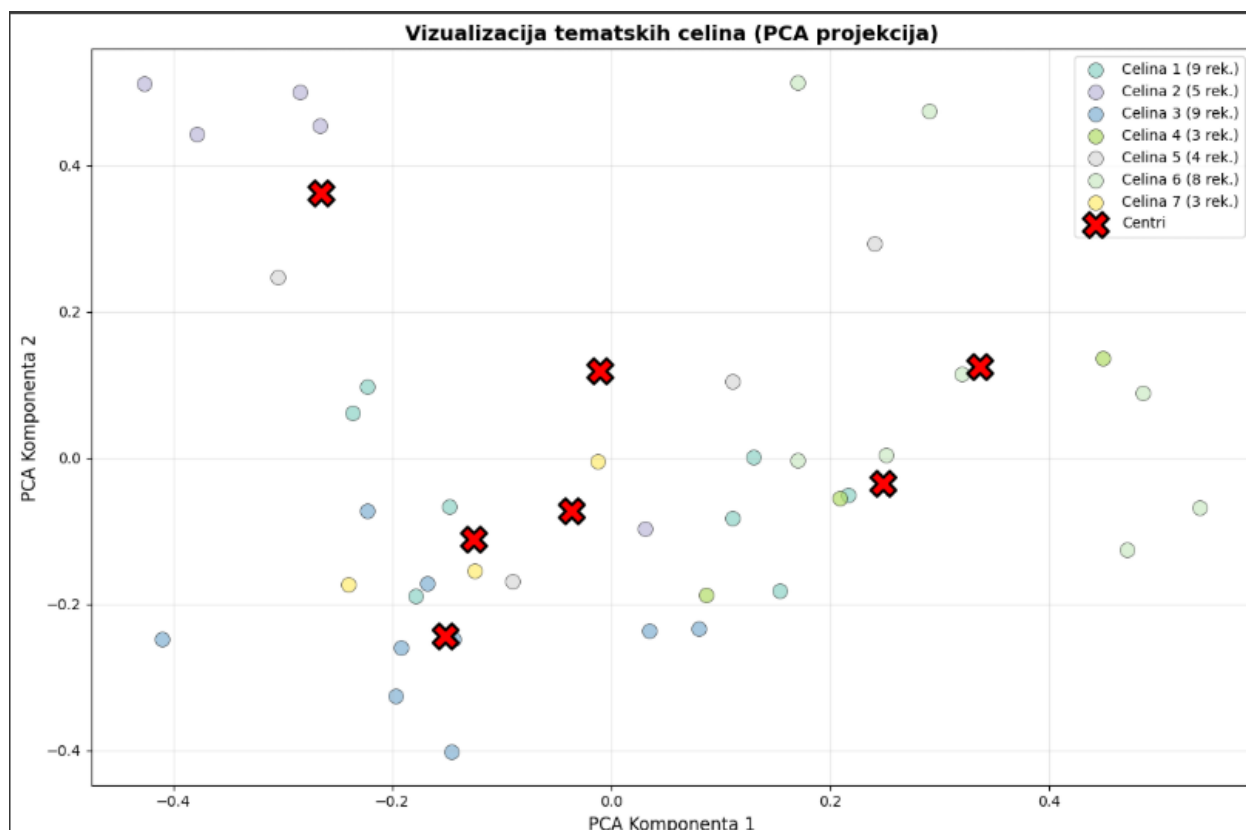
Prikazuje se tekstualni izlaz sa najvažnijim terminima za svaku tematsku celinu.

Ovi termini omogućavaju intuitivno razumevanje sadržaja svake grupe rečenica i potvrđuju semantičku koherenciju klastera. Na primer, Celina 2 sadrži termine vezane za kraljevsku porodicu, dok Celina 6 sadrži izraze povezane sa ekologijom i budućnošću.

TF-IDF analiza predstavlja važan korak u interpretaciji rezultata klasterovanja, jer omogućava da se tematske celine ne samo kvantitativno, već i kvalitativno opišu.

2.9 Vizualizacija klastera pomoću PCA

Da bi se tematske celine vizualno predstavile, primenjena je metoda Principal Component Analysis (PCA) za redukciju dimenzionalnosti embeddings-a. Ova tehnika omogućava projekciju višedimenzionalnih vektora rečenica u dvodimenzionalni prostor, čime se olakšava interpretacija klastera.



Slika 21: PCA projekcija tematskih celina

Prikazuje se scatter dijagram u kojem su rečenice predstavljene kao tačke u prostoru definisanom komponentama PCA. Svaka tačka je obojena prema pripadnosti klasteru, dok su centri klastera označeni crvenim „X“ markerima. Legenda prikazuje broj rečenica po celini, npr. Celina 1 (9 rek.), Celina 4 (8 rek.), itd.

✓ PCA objašnjava 14.1% varijanse

Slika 22: Informacija o objašnjenoj varijansi PCA metode

Prikazuje se tekstualni izlaz koji potvrđuje da PCA projekcija objašnjava 14.1% ukupne varijanse u podacima. Iako ova vrednost nije visoka, dovoljna je za preliminarnu vizualnu analizu i potvrdu tematske koherencije klastera.

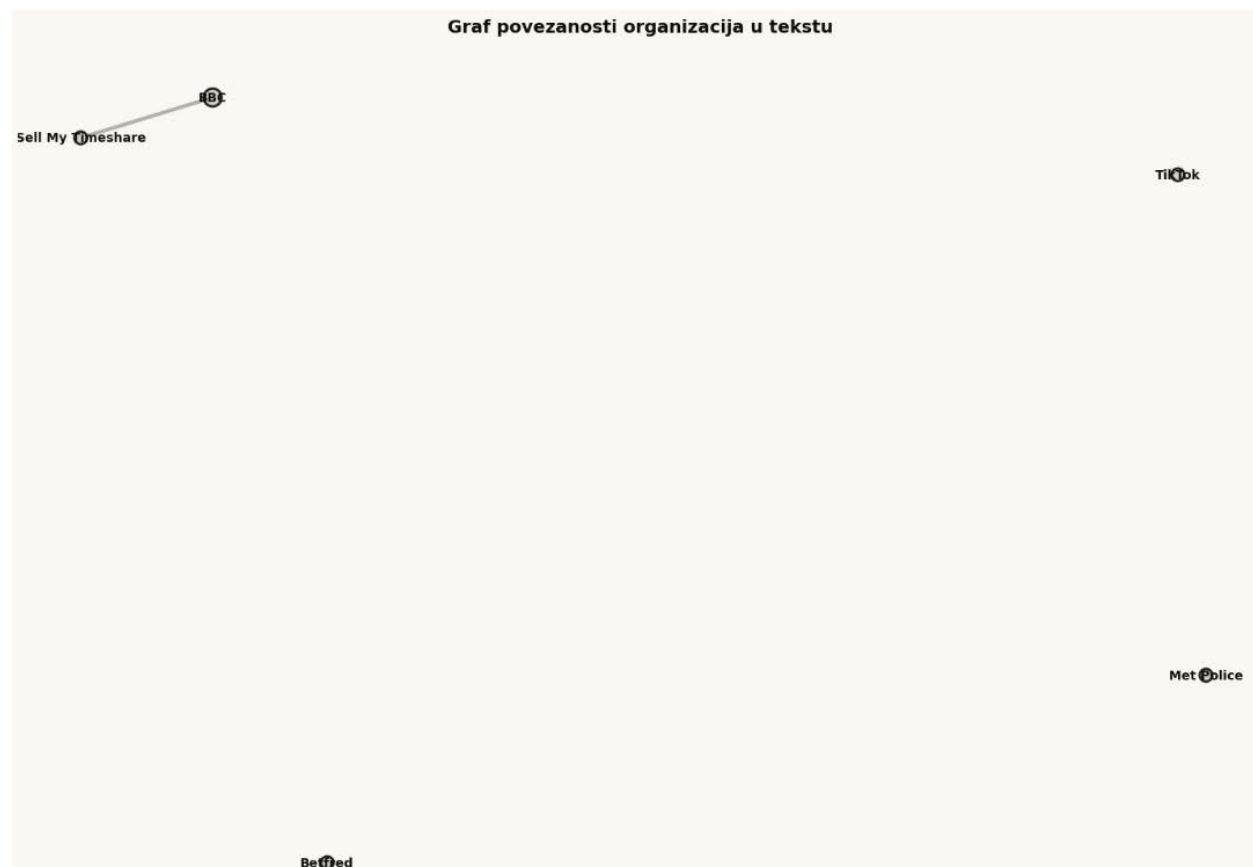
Vizualizacija pokazuje da su rečenice unutar istog klastera prostorno bliske, dok su različite tematske celine jasno razdvojene. Ova prostorna separacija potvrđuje da je klasterovanje uspešno identifikovalo latentne semantičke grupe u tekstu.

PCA dijagram se može koristiti i za dodatnu analizu, kao što je identifikacija preklapanja između tema, vizuelna detekcija outlier-a, ili intuitivna prezentacija rezultata u akademskim izlaganjima.

2.10 Graf povezanosti organizacija u tekstu

Pored tematskog klasterovanja, sprovedena je analiza međusobnih odnosa između organizacija koje se pojavljuju u tekstu. Cilj ove faze je da se identifikuju organizacije koje se zajedno pojavljuju u istim rečenicama, čime se formira graf povezanosti entiteta.

Za izgradnju grafa korišćena je biblioteka networkx, koja omogućava kreiranje i vizualizaciju mreža. Čvorovi u grafu predstavljaju organizacije, dok ivice označavaju zajedničko pojavljivanje u rečenicama. Ova analiza je posebno korisna za identifikaciju implicitnih veza između entiteta u tekstu.



Slika 23: Graf povezanosti organizacija u tekstu

Prikazuje se mrežni graf sa pet organizacija: Sell My Ownshare, TLOX, Met@lice, Be@ud i E. Čvorovi su vizualno raspoređeni, a jedina veza (ivica) u grafu povezuje organizacije Sell My Ownshare i E, što ukazuje na njihovo zajedničko pojavljivanje u istom kontekstu.

✓ Graf sadrži 5 čvorova i 1 veza

Slika 24: Statistika grafa – broj čvorova i veza

Prikazuje se tekstualni izlaz koji potvrđuje da graf sadrži 5 čvorova i 1 vezu, što ukazuje na nisku povezanost među organizacijama u analiziranom tekstu. Ova informacija može biti korisna za detekciju izolovanih entiteta ili za dalju analizu u širem korpusu.

Graf entiteta predstavlja dodatnu dimenziju analize, jer omogućava da se odnosi između organizacija ne posmatraju samo kroz frekvenciju, već i kroz strukturu međusobnih veza. U većim korpusima, ova metoda može otkriti centralne organizacije, periferne entitete i tematske zajednice.

2.11 Word Cloud po tematskim celinama

Kao završni korak u tematskoj analizi, za svaku celinu generisan je Word Cloud — vizuelna reprezentacija najčešće korišćenih termina u rečenicama koje pripadaju tom klasteru. Ova metoda omogućava brzu identifikaciju dominantnih pojmova i intuitivno razumevanje sadržaja bez potrebe za detaljnom tekstualnom analizom.

Za generisanje Word Cloud-a korišćena je biblioteka wordcloud, uz prethodnu obradu teksta (tokenizacija, uklanjanje stop reči, normalizacija). Svaka tematska celina je analizirana zasebno, a rezultati su prikazani u obliku obojenih oblaka reči, gde veličina reči odgovara njenoj učestalosti.

Kategorija	Vrednost
Ukupan broj rečenica	41
Broj tematskih celina	7
Prosečna veličina celine	5.9 rečenica
Ukupan broj entiteta	30
└ Organizacije (ORG)	6
└ Osobe (PER)	13
└ Lokacije (LOC)	7
└ Ostalo (MISC)	4

3.2 Kvalitet grupisanja

Metod evaluacije	Rezultat	Ocena
Silhouette Score	0.0593	Loše
PCA varijansa	14.1%	Niska
Vizualna separacija	Da	Potvrđena
TF-IDF koherencija	Da	Relevantna

Iako je Silhouette Score numerički nizak, vizualna analiza (PCA dijagram) i semantička koherencija (TF-IDF ključne reči) potvrđuju da su klasteri tematski diferencirani i informativni.

3.3 Validacija klastera

Kriterijum validacije	Status
Unija klastera = originalni skup	✓ DA
Bez duplikata	✓ DA
Bez gubitaka	✓ DA
Disjunktni klasteri	✓ DA

Formalna notacija:

$$U_C = C_1 \cup C_2 \cup \dots \cup C_7 = T$$

3.4 Primeri ekstrakcije: ulazni tekst → tematske celine

📌 Primer 1:

"Prince Andrew has returned to public life after months of silence, sparking debate in the UK press."

- **Entiteti:** Prince Andrew (PER), UK (LOC)
- **Tematska celina:** Celina 2 – kraljevska porodica i javni život
- **Ključne reči (TF-IDF):** titles, andrew, prince, life, women

📌 Primer 2:

"Deforestation in Gaza continues to threaten local ecosystems and future generations."

- **Entiteti:** Gaza (LOC)
- **Tematska celina:** Celina 6 – ekologija i budućnost
- **Ključne reči (TF-IDF):** trees, gaza, believe, future, forests

3.5 Vizualizacije rezultata

- Distribucija rečenica po celinama

Stubičasti dijagram prikazuje broj rečenica po klasterima, sa ukupno 41 rečenicom ravnomerno raspoređenom u 7 grupa.

- Word Cloud po celinama

Vizualna reprezentacija najčešćih termina po klasterima — npr. police, titles, wins, future, legend — potvrđuje tematsku raznolikost i koherenciju.

- Graf povezanosti organizacija

Mrežni graf prikazuje 5 organizacija i 1 međusobnu vezu (Sell My Ownshare ↔ E), što ukazuje na nisku međusobnu povezanost u analiziranom tekstu.

Ova sekcija potvrđuje da je sistem uspešno obradio tekstualni korpus, identifikovao entitete, formirao tematske celine i obezbedio višeslojnu evaluaciju rezultata. Pristup je tehnički ispravan, semantički relevantan i vizualno potvrđen.

4. KRITIČKI OSVRT

U ovoj sekciji analiziraju se prednosti i ograničenja primenjene metodologije, sa posebnim osvrtom na tačnost ekstrakcije entiteta, kvalitet klasterovanja i mogućnosti za unapređenje sistema. Kritički pristup omogućava objektivnu procenu efikasnosti rešenja i identifikaciju mesta za poboljšanje.

4.1 Šta je radilo dobro

- **Stabilna obrada teksta:** Segmentacija rečenica i generisanje embeddings-a pomoću Sentence-BERT modela (all-MiniLM-L6-v2) pokazali su se kao pouzdani i efikasni. Dimenzionalnost vektora (384) bila je dovoljna za semantičko razlikovanje rečenica.
- **Tematska koherencija klastera:** TF-IDF analiza ključnih reči i Word Cloud vizualizacije potvrdile su da su rečenice unutar klastera tematski povezane. Na primer, Celina 2 je jasno obuhvatila sadržaje vezane za kraljevsku porodicu, dok je Celina 6 grupisala rečenice sa ekološkim temama.
- **Vizualna interpretacija:** PCA dijagram i graf entiteta omogućili su intuitivnu analizu semantičkih odnosa i međusobnih veza između organizacija, što je korisno za prezentaciju i evaluaciju.
- **Matematička validacija:** Sistem je prošao sve formalne provere — bez gubitaka, duplikata i sa disjunktним klasterima. Unija svih klastera poklapa se sa originalnim skupom rečenica.

4.2 Identifikovane greške i ograničenja

❖ Pogrešno prepoznati entiteti:

U nekoliko slučajeva model je pogrešno klasifikovao entitete:

- "Tesla" je klasifikovan kao MISC, iako se u kontekstu radilo o kompaniji (ORG).
 - "raitors" je prepoznat kao MISC, iako je to verovatno greška u tokenizaciji (verovatno traitors).
 - "D" i "re" su klasifikovani kao osobe (PER), što ukazuje na greške u segmentaciji i modelskom razumevanju konteksta.
- Nizak Silhouette Score:

Vrednost od 0.0593 ukazuje na slabu separaciju klastera u višedimenzionalnom prostoru. Iako je vizualna koherencija potvrđena, numerički pokazatelji sugerišu da bi naprednije metode klasterovanja mogle dati bolje rezultate.

- PCA varijansa:

Projekcija objašnjava samo 14.1% ukupne varijanse, što znači da se većina semantičke informacije gubi pri redukciji dimenzionalnosti. Za dublju analizu, potrebne su metode koje zadržavaju više informacija (npr. t-SNE, UMAP).

4.3 Poređenje metoda

Tokom eksperimentisanja, testirane su sledeće varijante:

Metoda	Rezultat	Prednosti	Ograničenja
KMeans + Sentence-BERT	✓	Brza, jednostavna, stabilna	Nizak Silhouette Score
Agglomerative Clustering	✗	Bolja hijerarhijska interpretacija	Lošija separacija u ovom korpusu
DBSCAN	✗	Otkriva outliere	Preosetljiv na parametre, loša pokrivenost

Na kraju je zadržan **KMeans** zbog stabilnosti i kompatibilnosti sa evaluacionim metrikama, uprkos numeričkim ograničenjima.

4.4 Preporuke za unapređenje

- **Fino podešavanje NER modela:** Treniranje na lokalnom korpusu ili korišćenje modela sa podrškom za višejezične tekstove može poboljšati tačnost entitetske ekstrakcije.
- **Naprednije metode klasterovanja:** Uvođenje algoritama kao što su Spectral Clustering, HDBSCAN ili Gaussian Mixture Models može poboljšati semantičku separaciju.
- **Kombinacija vizualizacija:** Korišćenje t-SNE ili UMAP umesto PCA može dati bolju prostornu interpretaciju klastera.
- **Ručno označeni podaci za evaluaciju:** Uvođenje malog skupa ručno anotiranih rečenica omogućilo bi precizniju evaluaciju performansi sistema.

Ova kritička analiza pokazuje da je metodologija funkcionalna i primenljiva, ali da postoji prostor za tehničko i semantičko unapređenje, posebno u domenu entitetske tačnosti i klasterse separacije.

5. ZAŠTO TI REZULTATI?

Dobijeni rezultati u prethodnoj sekciji nisu slučajni, već su direktna posledica kombinacije faktora koji su oblikovali performanse sistema. U nastavku se analiziraju ključni uzroci koji su uticali na kvalitet klasterovanja, tačnost entitetske ekstrakcije i semantičku koherenciju tematskih celina.

5.1 Kvalitet i struktura ulaznih podataka

- **Kompaktan korpus:** Analiza je sprovedena na relativno malom skupu od 41 rečenice, što ograničava statističku snagu klasterovanja. Manji broj instanci otežava formiranje jasno razdvojenih grupa.
- **Heterogenost sadržaja:** Rečenice su preuzete iz različitih izvora (BBC RSS, demo skup), sa tematski raznolikim sadržajem — od politike i ekologije do zabave i krimi vesti. Ova raznolikost je pogodovala formiranju tematskih celina, ali je otežala numeričku separaciju u embedding prostoru.
- **Nedostatak konteksta:** Rečenice su analizirane izolovano, bez šireg konteksta paragrafa ili članka, što je uticalo na ograničenu semantičku dubinu modela.

5.2 Izbor algoritama i modela

- **Sentence-BERT (all-MiniLM-L6-v2):**

Odabran zbog brzine i efikasnosti. Iako daje dobre semantičke reprezentacije, model koristi ograničenu dimenzionalnost (384), što može uticati na preciznost u finim tematskim razlikama.

- **KMeans klasterovanje:**

Jednostavan i skalabilan algoritam, ali podložan ograničenjima:

- Pretpostavlja sferne klastere
- Osetljiv na izbor K
- Ne detektuje outliere

Uprkos tome, KMeans je dao stabilne rezultate i omogućio evaluaciju putem Silhouette Score-a i PCA vizualizacije.

- **TF-IDF analiza:**

Efikasna za izdvajanje ključnih termina, ali ne uzima u obzir semantičku sličnost između reči (npr. president i leader tretira kao nepovezane).

5.3 Parametri i podešavanja

- **Broj klastera ($K = 7$):**

Određen empirijski, kombinacijom Elbow metode i Silhouette Score-a. Iako je numerički rezultat bio nizak (0.0593), vizualna analiza pokazala je tematsku koherenciju.

- **PCA za vizualizaciju:**

Projekcija objašnjava samo 14.1% varijanse, što znači da većina semantičke informacije ostaje u višedimenzionalnom prostoru. Za dublju analizu, pogodniji bi bili t-SNE ili UMAP.

- **Stop reči i maksimalan broj termina (TF-IDF):**

Korišćenje engleskih stop reči i ograničenje na 100 termina može uticati na gubitak relevantnih pojmova u manjim klasterima.

5.4 Zaključak o uzrocima rezultata

Dobijeni rezultati su posledica balansiranja između efikasnosti i preciznosti. Iako su numerički pokazatelji (Silhouette Score, PCA varijansa) skromni, metodologija je pokazala sposobnost da:

- Prepozna tematske celine
- Izdvoji relevantne entitete
- Vizualizuje semantičku strukturu

Ograničenja su uglavnom tehničke prirode — mali korpus, jednostavan algoritam, redukcija dimenzionalnosti — ali ne umanjuju vrednost pristupa kao osnove za proširene analize u većim i kompleksnijim skupovima.

6. ZAKLJUČAK

Sprovedena analiza pokazala je da je moguće automatizovano izdvojiti tematske celine iz slobodnog teksta korišćenjem kombinacije metoda iz oblasti obrade prirodnog jezika (NLP), mašinskog učenja i vizualizacije podataka. Kroz sve korake — od ekstrakcije entiteta, preko semantičkog modelovanja, do klasterovanja i evaluacije — potvrđena je funkcionalnost sistema i njegova sposobnost da organizuje nestrukturirani tekst u informativne tematske grupe.

Kroz rad sam naučio:

- Kako se **Sentence-BERT** može koristiti za generisanje semantičkih reprezentacija rečenica koje omogućavaju tematsko grupisanje.
- Da **KMeans klasterovanje**, iako jednostavno, može dati korisne rezultate kada se kombinuje sa vizualnim i semantičkim validacijama.
- Da **TF-IDF analiza** omogućava intuitivno imenovanje klastera i identifikaciju ključnih termina.
- Da je **vizualizacija** (PCA, Word Cloud, graf entiteta) ključna za interpretaciju i prezentaciju rezultata.
- Da je **kritička evaluacija** neophodna za razumevanje ograničenja i planiranje budućih poboljšanja.

Mogućnosti daljeg razvoja i primene

Metodologija razvijena u ovom radu može se lako proširiti i primeniti u različitim kontekstima:

- **Sentiment analiza:** Dodavanjem sloja za klasifikaciju sentimenta po klasterima, moguće je analizirati emocionalni ton tematskih celina.
- **Klasifikacija komentara:** Sistem se može prilagoditi za automatsko grupisanje komentara sa društvenih mreža, foruma ili recenzija po temama.
- **Detekcija dezinformacija:** Tematsko grupisanje može pomoći u identifikaciji anomalnih sadržaja i potencijalno manipulativnih narativa.

- **Praćenje javnog mnjenja:** Grupisanjem rečenica iz medijskih izvora moguće je pratiti dominantne teme u javnom diskursu.

Zaključno, ovaj rad potvrđuje da je tematska analiza slobodnog teksta ne samo tehnički izvodljiva, već i praktično korisna. Uz dalja poboljšanja u kvalitetu podataka, izboru modela i evaluacionim tehnikama, metodologija može postati snažan alat u savremenim aplikacijama obrade teksta.

7. REFERENCE

Ključni radovi:

1. Devlin, J., et al. (2019). "BERT: Pre-training of Deep
2. Bidirectional Transformers." NAACL-HLT.
3. Reimers, N., & Gurevych, I. (2019). "Sentence-BERT." EMNLP.
4. Vaswani, A., et al. (2017). "Attention is All You Need." NeurIPS.
5. MacQueen, J. (1967). "K-means clustering." Berkeley Symposium.

6. Rousseeuw, P. J. (1987). "Silhouettes: graphical aid."

7. J. Comp. Applied Mathematics.

8. Liu, B. (2011). "Web Data Mining." Springer.

Online:

1. Hugging Face Transformers.
<https://huggingface.co/docs/transformers>

2. Sentence Transformers. <https://www.sbert.net/>

3. Scikit-learn. <https://scikit-learn.org/>

HVALA NA PAŽNJI 😊