SocioScope - Interim Presentation

Ofir Fichman, Pavel Fadeev, Israel Peled

Project Review

- Developing a real-time "Public Pulse" system that monitors Twitter to quantify public sentiment across 10 key socio-economic pillars.
- Transitioned from manual annotation to an automated Synthetic Data Pipeline using LLMs to generate a high-quality, 10,000-sample dataset.
- Applying advanced ABSA to the socio-economic domain (beyond standard product reviews) by outputting a 10-dimensional sentiment vector per tweet.

Why Use SocioScope?

+1
(Government Support)
-1
(Cost of Living Concern)

Nuanced Understanding:

Moving beyond simple 'Positive/Negative' labels to capture complex public opinion.

For Example:
A citizen might support the Government (+1) while simultaneously criticizing the Cost of Living (-1) in the same tweet.

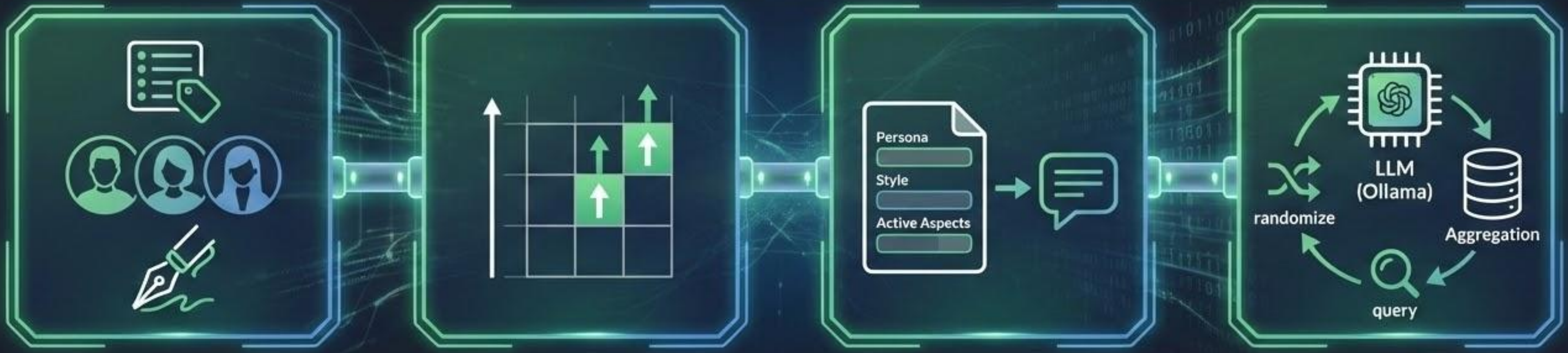
Empowering Policymakers:

Providing mayors and government officials with precise analytics to understand exactly what troubles the public, enabling data-driven decision-making.

Previous work

Title / Year	The Task	Methods	Data	Results	Relation to SocioScope
MEMD-ABSA: A Multi-Element Multi-Domain Dataset for Aspect-Based Sentiment Analysis (2023)	Extracting the Aspect, Category, Opinion, and Sentiment even when they are not explicitly mentioned.	Generative Baselines (BART/T5) + Multi-Domain Training	20,000 sentences across 5 domains (Books, Clothing, etc.)	Revealed that mining **implicit** aspects and opinions remains the biggest challenge in open-domain ABSA.	**Directly validates our use of LLMs** to infer implicit socio-economic sentiment from vague tweets.
Aspect-Based Sentiment Analysis Using BERT(2019)	**Advanced Modeling:** Using pre-trained BERT to identify aspects and sentiments	Fine-tuned BERT + Sentence-Pair Modeling	SemEval-2015 & SemEval-2016	Outperformed previous SVM/CRF baselines	Justifies our use of BERT-like Transformers and "Supervised" synthetic labels.
Sentiment Analysis in the Era of LLMs: A Reality Check(2023)	**Evaluating LLMs vs. Small Models**	Comparing Zero-shot LLMs (ChatGPT) vs. Fine-tuned models (BERT)	26 Standard sentiment datasets (IMDB, Twitter, Rest14)	Fine-tuned small models (like BERT) often outperform Zero-shot LLMs in specific tasks.	Supports our hypothesis that a fine-tuned BERT can surpass Gemma's zero-shot performance.

Data Generation Pipeline



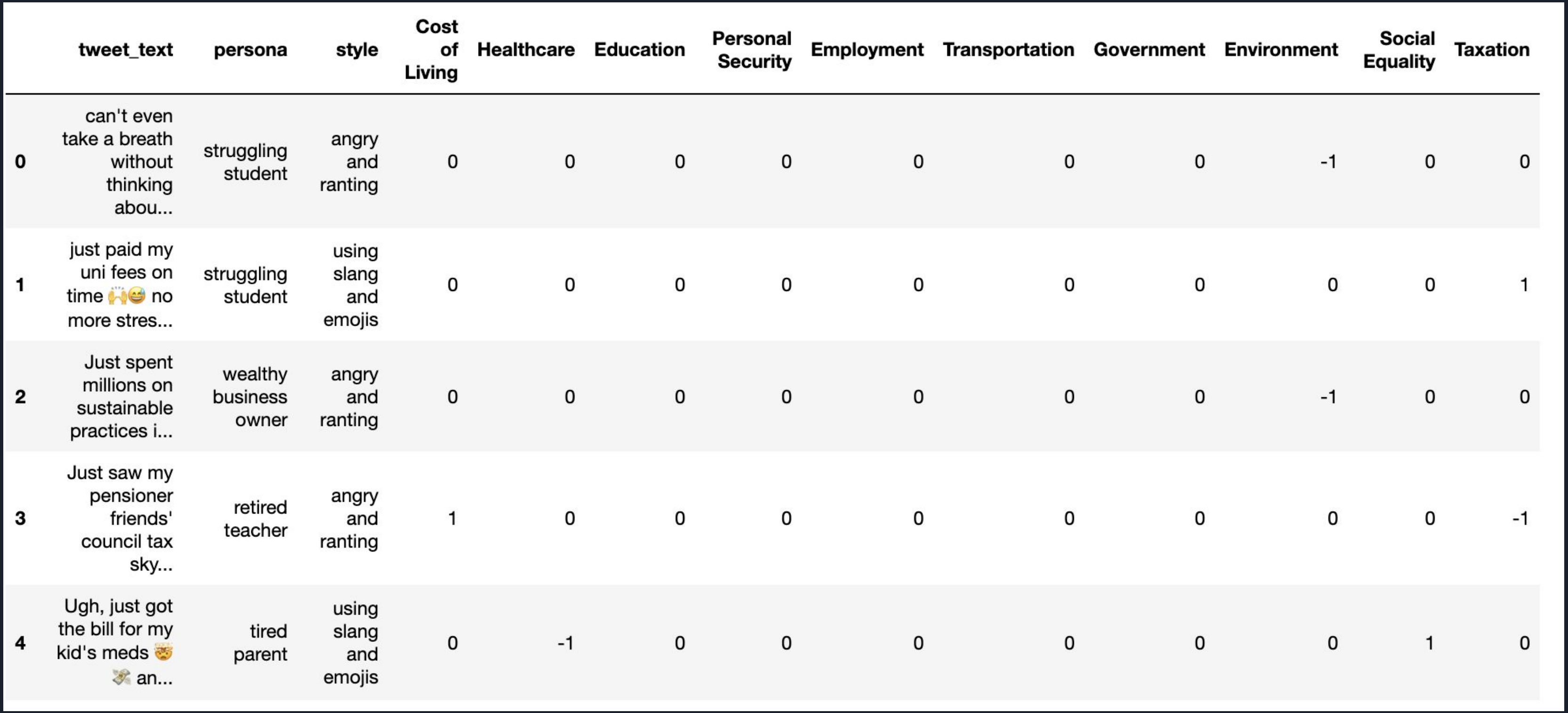
1. Define Attributes
Defined a list of aspects, personas, and writing styles for the model to generate tweets from.

2. Generate Target Vector
Created a function that generates a target vector, selecting 1-2 active aspects and assigning them non-zero sentiment.

3. Construct Dynamic Prompts
Constructed dynamic prompts that instruct the model to adopt a persona and write about the active aspects in a specific style.

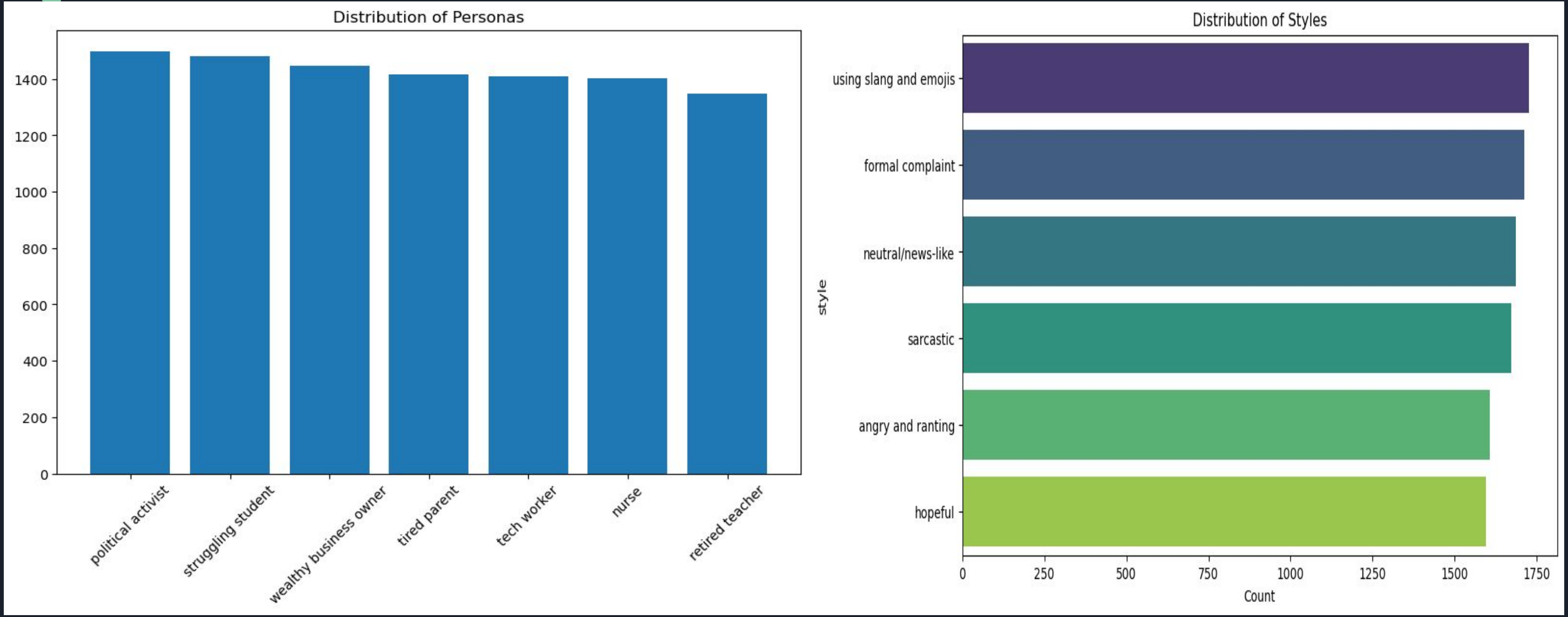
4. Main Loop & LLM Query
Implemented the main loop to orchestrate the pipeline: it randomizes attributes, queries the LLM via Ollama, and aggregates the results.

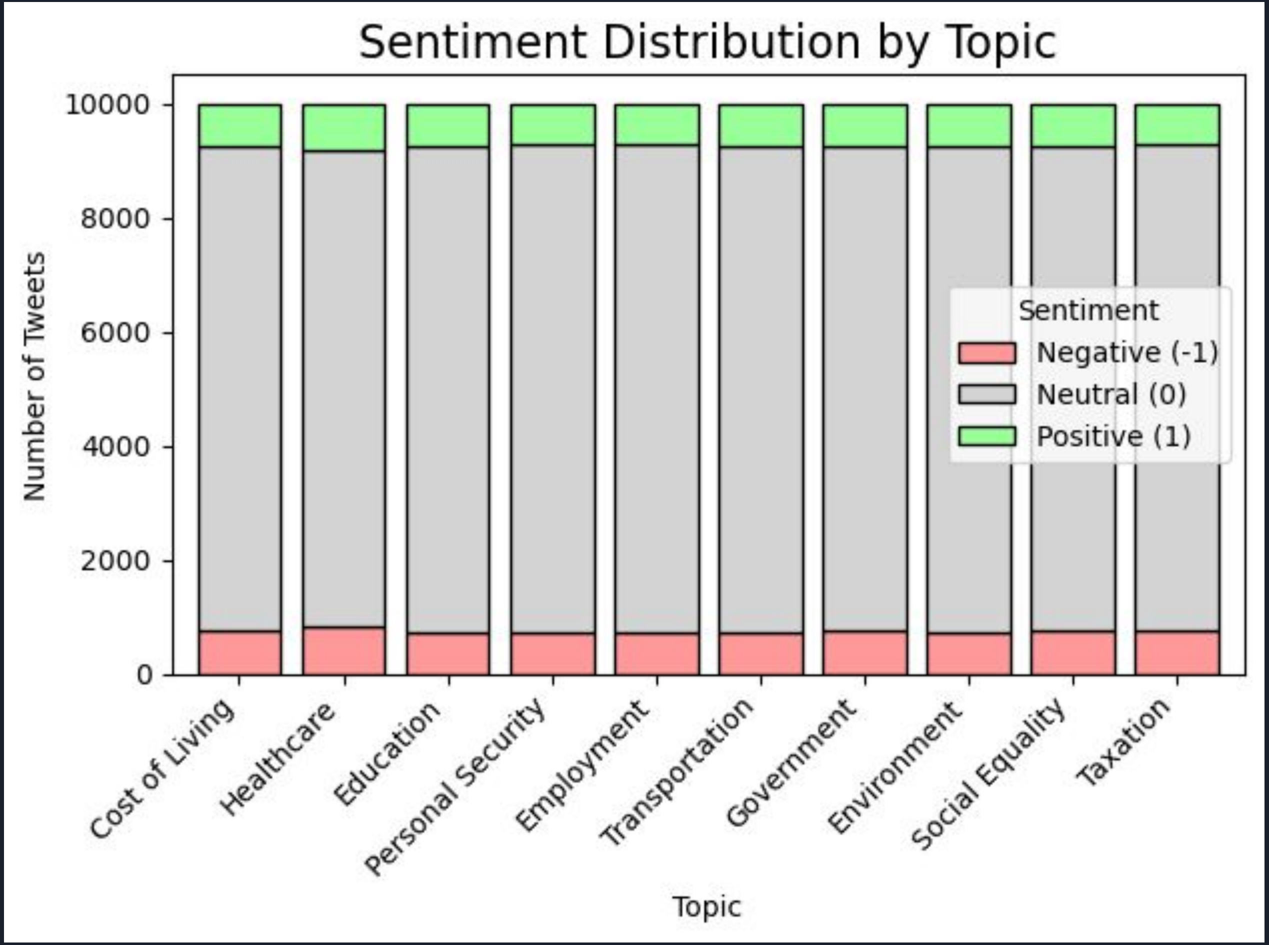
Dataset



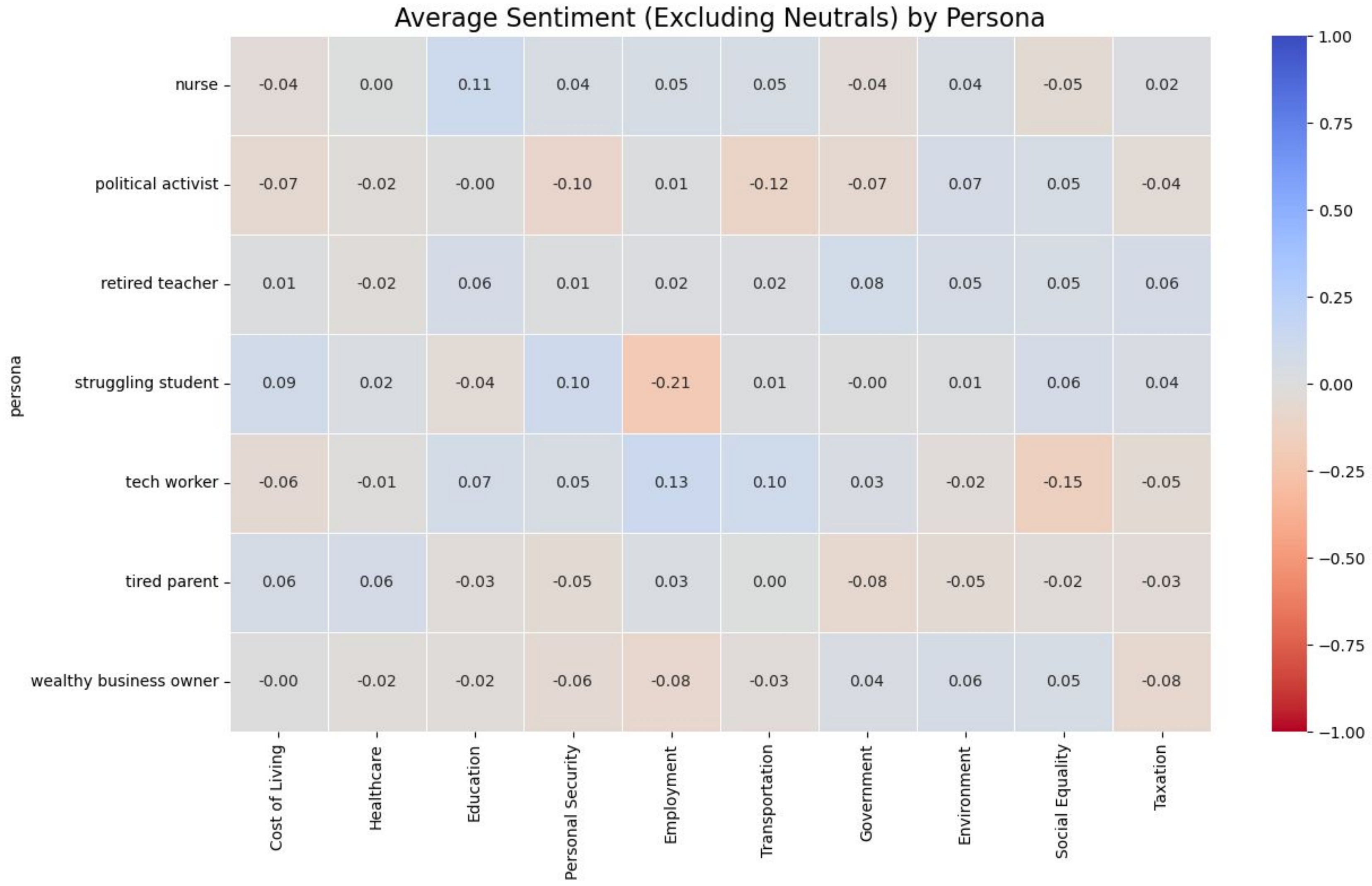
	tweet_text	persona	style	Cost of Living	Healthcare	Education	Personal Security	Employment	Transportation	Government	Environment	Social Equality	Taxation
0	can't even take a breath without thinking abou...	struggling student	angry and ranting	0	0	0	0	0	0	0	-1	0	0
1	just paid my uni fees on time 🙌😅 no more stres...	struggling student	using slang and emojis	0	0	0	0	0	0	0	0	0	1
2	Just spent millions on sustainable practices i...	wealthy business owner	angry and ranting	0	0	0	0	0	0	0	-1	0	0
3	Just saw my pensioner friends' council tax sky...	retired teacher	angry and ranting	1	0	0	0	0	0	0	0	0	-1
4	Ugh, just got the bill for my kid's meds 🤯 💸 an...	tired parent	using slang and emojis	0	-1	0	0	0	0	0	0	1	0

Distributions

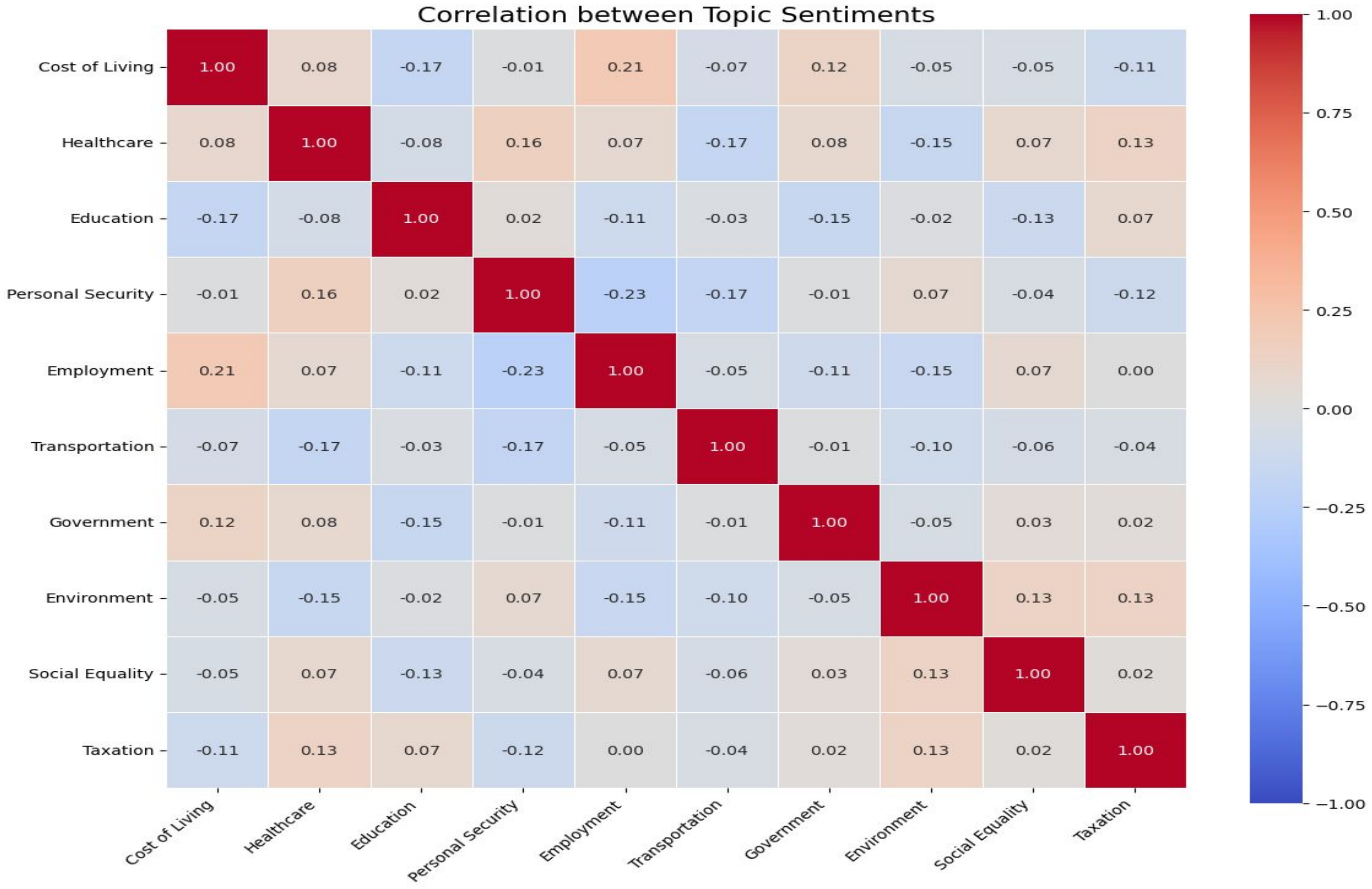




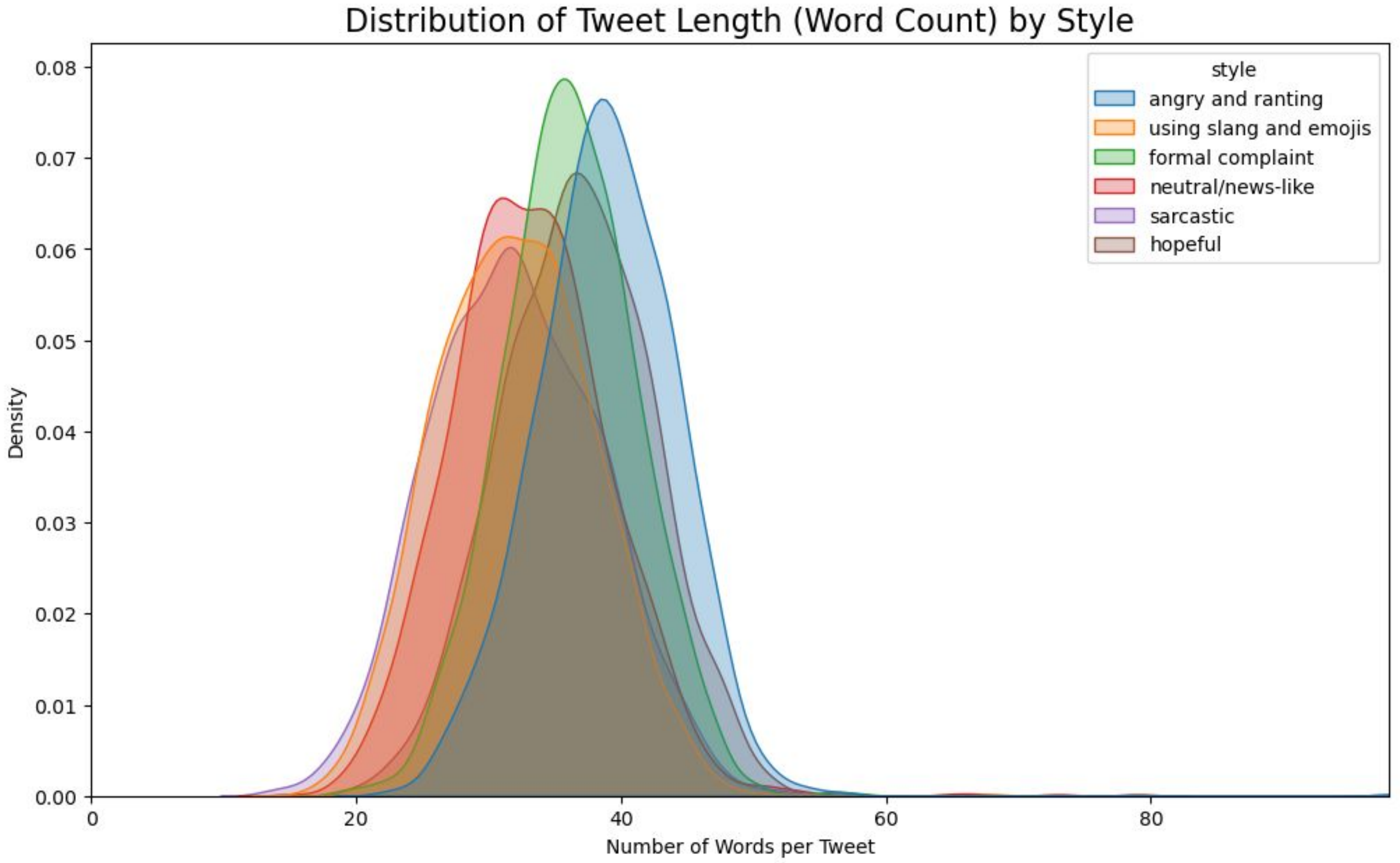
Sentiment Distribution by Topic
Number of Tweets
10000
8000
6000
4000
2000
0
Cost of Living
Healthcare
Education
Personal Security
Employment
Transportation
Government
Environment
Social Equality
Taxation
Topic
Sentiment
Negative (-1)
Neutral (0)
Positive (1)



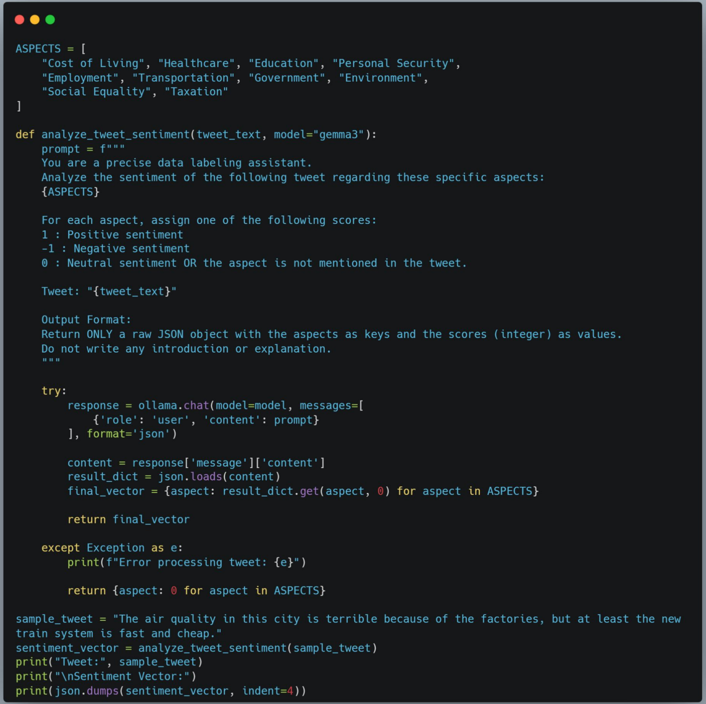
persona	Cost of Living	Healthcare	Education	Personal Security	Employment	Transportation	Government	Environment	Social Equality	Taxation
nurse	-0.04	0.00	0.11	0.04	0.05	0.05	-0.04	0.04	-0.05	0.02
political activist	-0.07	-0.02	-0.00	-0.10	0.01	-0.12	-0.07	0.07	0.05	-0.04
retired teacher	0.01	-0.02	0.06	0.01	0.02	0.02	0.08	0.05	0.05	0.06
struggling student	0.09	0.02	-0.04	0.10	-0.21	0.01	-0.00	0.01	0.06	0.04
tech worker	-0.06	-0.01	0.07	0.05	0.13	0.10	0.03	-0.02	-0.15	-0.05
tired parent	0.06	0.06	-0.03	-0.05	0.03	0.00	-0.08	-0.05	-0.02	-0.03
wealthy business owner	-0.00	-0.02	-0.02	-0.06	-0.08	-0.03	0.04	0.06	0.05	-0.08



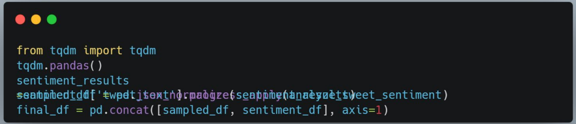
	Cost of Living	Healthcare	Education	Personal Security	Employment	Transportation	Government	Environment	Social Equality	Taxation
Cost of Living	1.00	0.08	-0.17	-0.01	0.21	-0.07	0.12	-0.05	-0.05	-0.11
Healthcare	0.08	1.00	-0.08	0.16	0.07	-0.17	0.08	-0.15	0.07	0.13
Education	-0.17	-0.08	1.00	0.02	-0.11	-0.03	-0.15	-0.02	-0.13	0.07
Personal Security	-0.01	0.16	0.02	1.00	-0.23	-0.17	-0.01	0.07	-0.04	-0.12
Employment	0.21	0.07	-0.11	-0.23	1.00	-0.05	-0.11	-0.15	0.07	0.00
Transportation	-0.07	-0.17	-0.03	-0.17	-0.05	1.00	-0.01	-0.10	-0.06	-0.04
Government	0.12	0.08	-0.15	-0.01	-0.11	-0.01	1.00	-0.05	0.03	0.02
Environment	-0.05	-0.15	-0.02	0.07	-0.15	-0.10	-0.05	1.00	0.13	0.13
Social Equality	-0.05	0.07	-0.13	-0.04	0.07	-0.06	0.03	0.13	1.00	0.02
Taxation	-0.11	0.13	0.07	-0.12	0.00	-0.04	0.02	0.13	0.02	1.00


Distribution of Tweet Length (Word Count) by Style
style
angry and ranting
using slang and emojis
formal complaint
neutral/news-like
sarcastic
hopeful
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0
20
40
60
80
Number of Words per Tweet

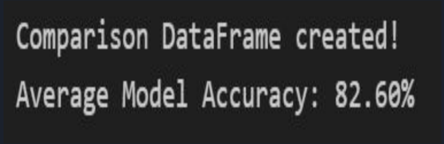
Baseline



```python
ASPECTS = [
    "Cost of Living", "Healthcare", "Education", "Personal Security",
    "Employment", "Transportation", "Government", "Environment",
    "Social Equality", "Taxation"
]

def analyze_tweet_sentiment(tweet_text, model="gemma3"):
    prompt = f"""
    You are a precise data labeling assistant.
    Analyze the sentiment of the following tweet regarding these specific aspects:
    {ASPECTS}

    For each aspect, assign one of the following scores:
    1 : Positive sentiment
    -1 : Negative sentiment
    0 : Neutral sentiment OR the aspect is not mentioned in the tweet.

    Tweet: "{tweet_text}"

    Output Format:
    Return ONLY a raw JSON object with the aspects as keys and the scores (integer) as values.
    Do not write any introduction or explanation.
    """

    try:
        response = ollama.chat(model=model, messages=[
            {'role': 'user', 'content': prompt}
        ], format='json')

        content = response['message']['content']
        result_dict = json.loads(content)
        final_vector = {aspect: result_dict.get(aspect, 0) for aspect in ASPECTS}

        return final_vector

    except Exception as e:
        print(f"Error processing tweet: {e}")

        return {aspect: 0 for aspect in ASPECTS}

sample_tweet = "The air quality in this city is terrible because of the factories, but at least the new
train system is fast and cheap."
sentiment_vector = analyze_tweet_sentiment(sample_tweet)
print("Tweet:", sample_tweet)
print("\nSentiment Vector:")
print(json.dumps(sentiment_vector, indent=4))
```



```python
from tqdm import tqdm
tqdm.pandas()
sentiment_results
= sampled_df['tweet_text'].progress_apply(analyze_tweet_sentiment)
sentiment_df = pd.json_normalize(sentiment_results)
final_df = pd.concat([sampled_df, sentiment_df], axis=1)
```



```
Comparison DataFrame created!
Average Model Accuracy: 82.60%
```

Is our work done? Absolutely not.

Current Objectives:

Logical Validation: Verify consistency between sentiment score vectors and tweet content to detect logical contradictions.

Model Optimization: Surpass current metrics by training a BERT model.

Data Augmentation: Expand the dataset through web scraping of tweets.