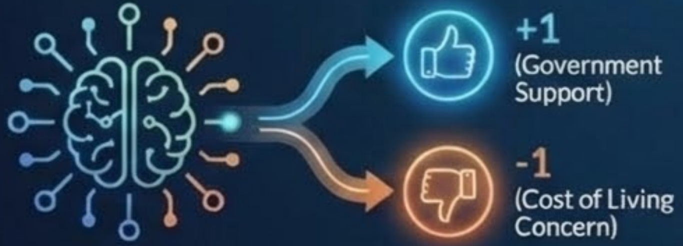
SocioScope - Interim Presentation

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?



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.

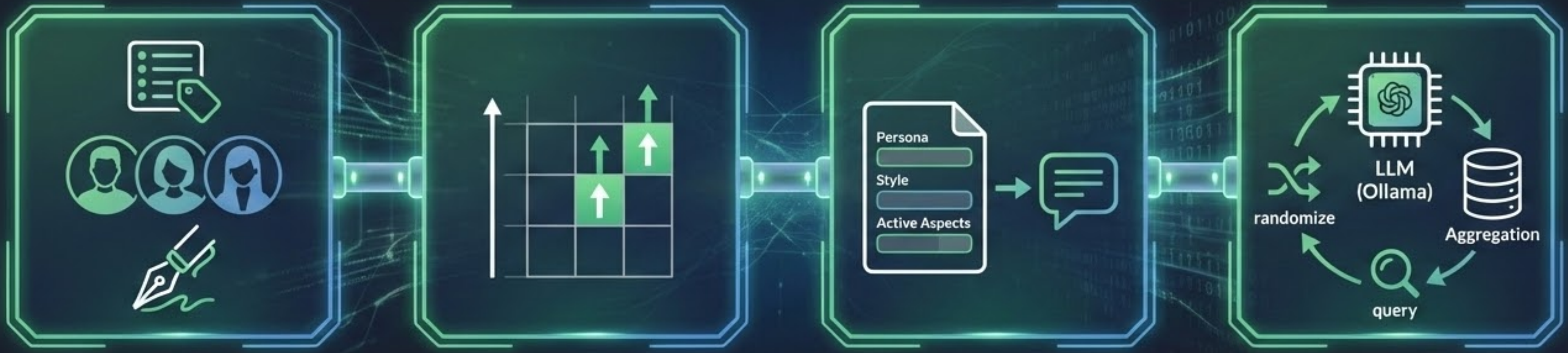
Empowering Policymakers:

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.

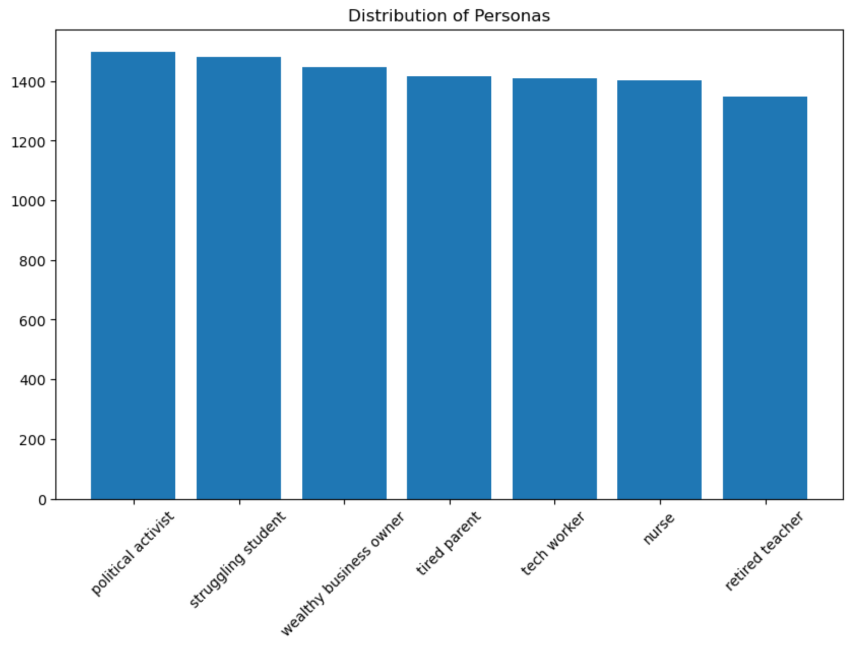
3. Construct Dynamic Prompts
Constructed dynamic prompts that instruct the model to adopt a persona and write about the active aspects in a specific style.

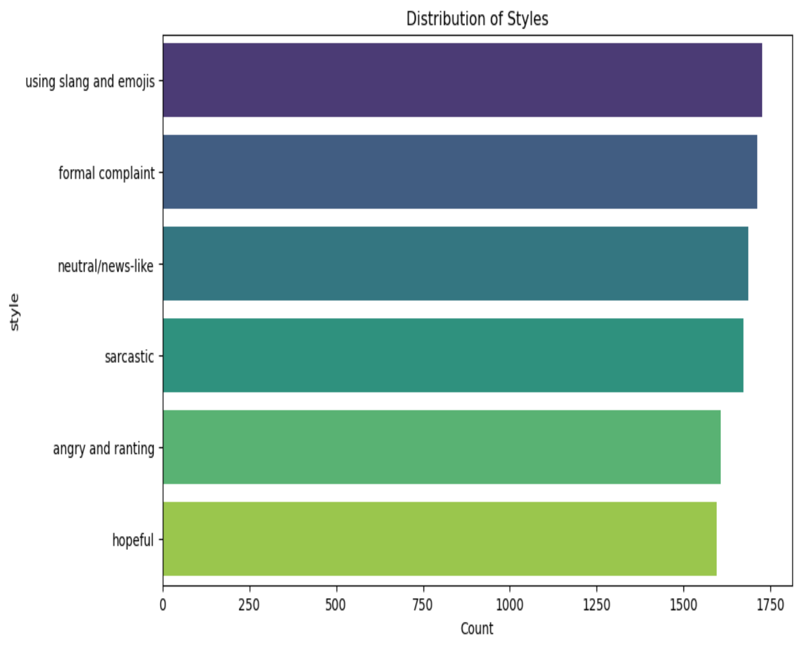
4. Main Loop & LLM Query
Implemented the main loop to orchestrate the pipeline: it randomizes attributes, queries the LLM via Ollama, and aggregates the results.

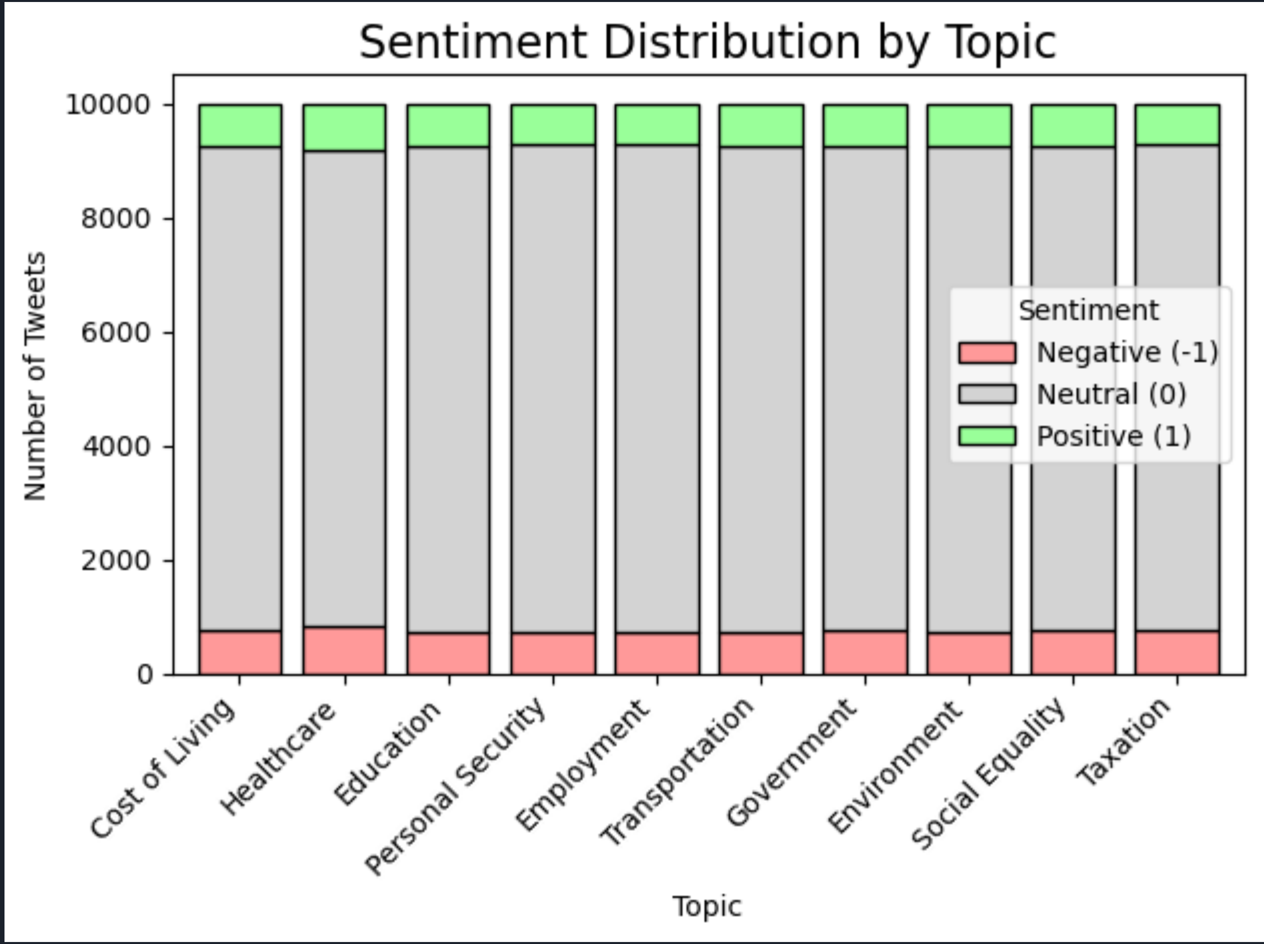
Dataset

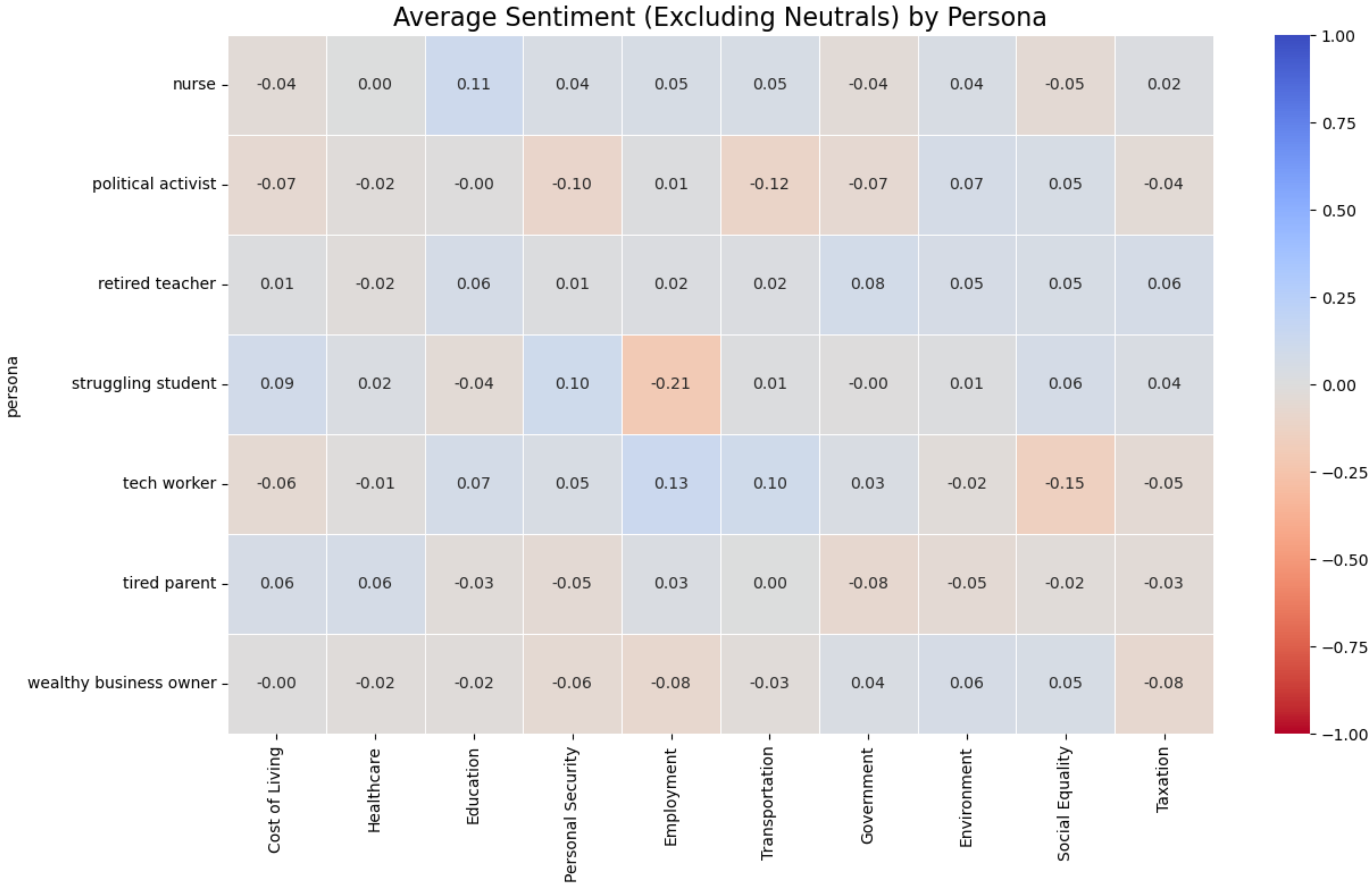
	tweet_text	persona	style	Cost of Living	Healthcare	Education	Personal Security	Employment	Transportation	Government	Environment	Social Equality	Taxation
0	can't even take a breath without thinking abou...	struggling student	angry and ranting	0	0	0	0	0	0	0	-1	0	0
1	just paid my uni fees on time 🙌😅 no 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

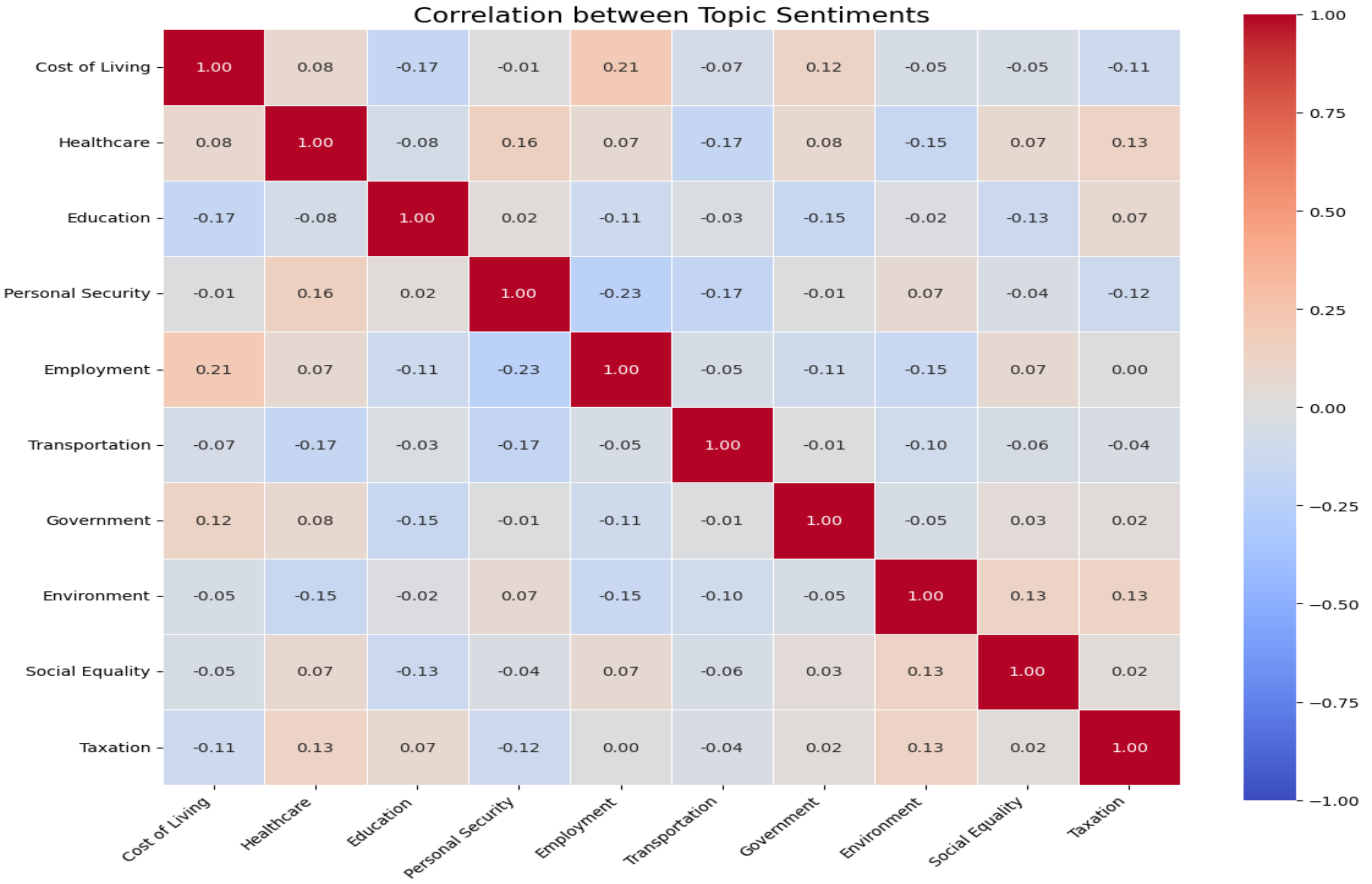
Distributions

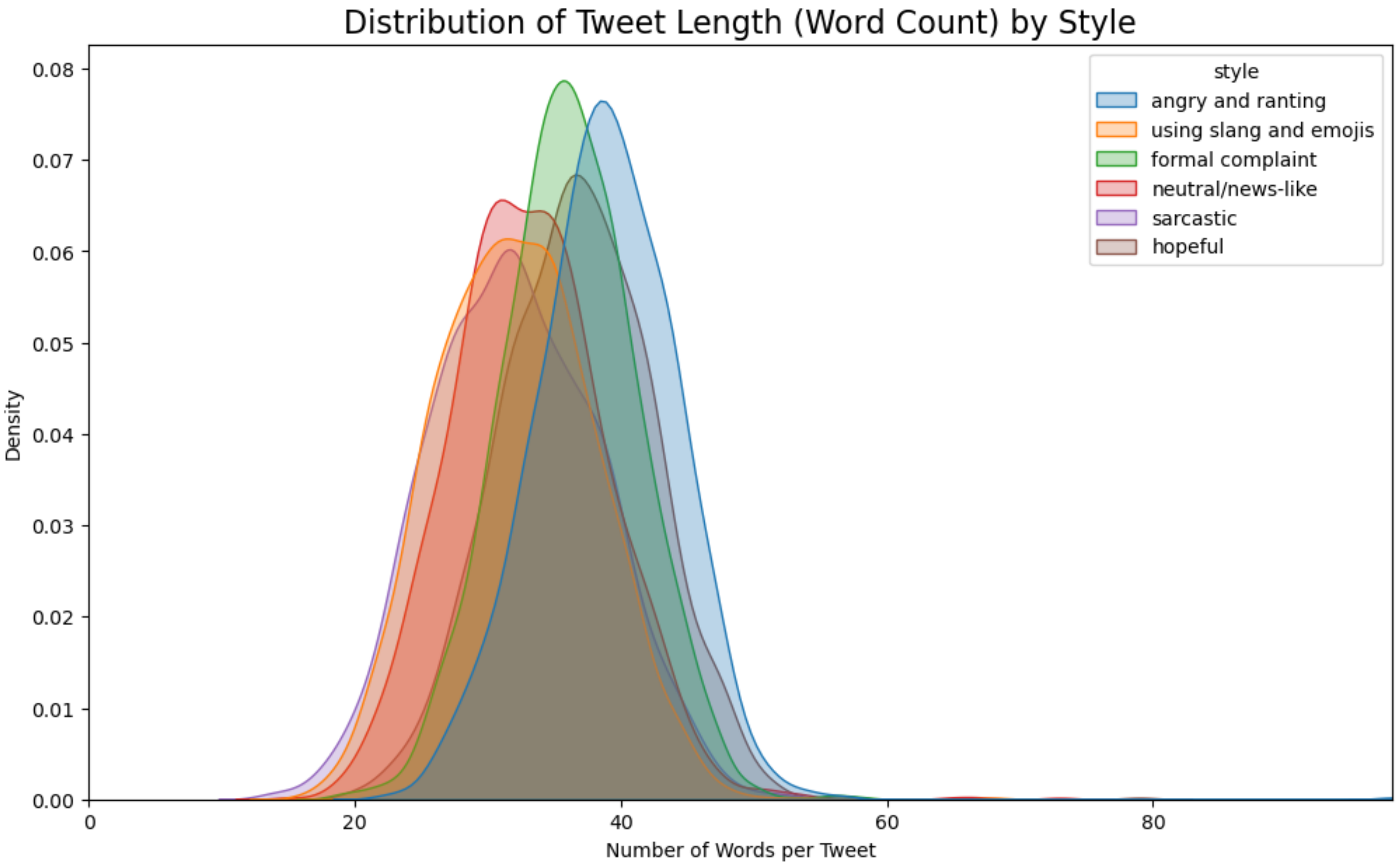





Sentiment Distribution by Topic
Number of Tweets
10000
8000
6000
4000
2000
0
Sentiment
Negative (-1)
Neutral (0)
Positive (1)
Cost of Living
Healthcare
Education
Personal Security
Employment
Transportation
Government
Environment
Social Equality
Taxation
Topic


Average Sentiment (Excluding Neutrals) by Persona
persona
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
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Correlation between Topic Sentiments
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
Cost of Living
Healthcare
Education
Personal Security
Employment
Transportation
Government
Environment
Social Equality
Taxation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−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
def build_prompt(tweet_text: str) -> str:

    aspects_str = "\n".join(f"- {a}" for a in ASPECTS)
    prompt = f"""
You are an assistant for aspect-based sentiment analysis.

Given a social-media post, you must assign a sentiment score (-1, 0, +1)
for each of the following aspects:

{aspects_str}

Meaning of scores:
-1 = clearly negative sentiment toward this aspect
 0 = not mentioned or neutral
+1 = clearly positive sentiment toward this aspect

Post:
\"\"\"{tweet_text}\"\"\"

Return ONLY valid JSON with keys exactly the aspect names and integer values -1, 0, or +1.
Example format:
{{
  "Cost of Living": -1,
  "Healthcare": 0,
  ...
}}
"""
    return prompt.strip()
```

```python
def predict_on_dataset(data: List[Dict[str, Any]], model_name: str):

    N = len(data)
    A = len(ASPECTS)

    y_true_detect = np.zeros((N, A), dtype=int)
    y_true_sign = np.zeros((N, A), dtype=int)
    y_pred_detect = np.zeros((N, A), dtype=int)
    y_pred_sign = np.zeros((N, A), dtype=int)

    for i, row in enumerate(tqdm(data, desc="Predicting with Gemma")):
        text = row["tweet_text"]
        labels = row["labels"]  # dict: aspect -> -1/0/+1

        # y_true
        for j, asp in enumerate(ASPECTS):
            v = int(labels.get(asp, 0))
            y_true_sign[i, j] = v
            if v != 0:
                y_true_detect[i, j] = 1

        # запрос к Gemma
        prompt = build_prompt(text)
        pred_dict = call_gemma(model_name, prompt)

        # y_pred
        for j, asp in enumerate(ASPECTS):
            raw_v = pred_dict.get(asp, 0)
            v = parse_label(raw_v)
            if v != 0:
                y_pred_detect[i, j] = 1
                y_pred_sign[i, j] = v
            else:
                y_pred_detect[i, j] = 0
                y_pred_sign[i, j] = 0

    return y_true_detect, y_true_sign, y_pred_detect, y_pred_sign
```

Aspect	det_precision	det_recall	det_f1	det_accuracy	sign_precision_macro	sign_recall_macro	sign_f1_macro	sign_accuracy
Cost of Living	0.462	0.885	0.607	0.807	0.561	0.620	0.512	0.749
Healthcare	0.568	0.952	0.711	0.872	0.630	0.779	0.666	0.834
Education	0.500	0.962	0.658	0.857	0.580	0.755	0.626	0.821
Personal Security	0.799	0.886	0.840	0.957	0.773	0.786	0.766	0.933
Employment	0.558	0.876	0.682	0.875	0.641	0.778	0.688	0.853
Transportation	0.935	0.938	0.937	0.982	0.835	0.823	0.821	0.953
Government	0.342	0.965	0.505	0.707	0.573	0.732	0.568	0.680
Environment	0.918	0.880	0.899	0.974	0.817	0.776	0.779	0.945
Social Equality	0.625	0.863	0.725	0.904	0.595	0.679	0.627	0.860
Taxation	0.780	0.913	0.841	0.949	0.762	0.779	0.741	0.914

Aspect	sign_precision_macro	sign_recall_macro	sign_f1_macro	sign_accuracy
Cost of Living	0.561	0.620	0.512	0.749
Healthcare	0.630	0.779	0.666	0.834
Education	0.580	0.755	0.626	0.821
Personal Security	0.773	0.786	0.766	0.933
Employment	0.641	0.778	0.688	0.853
Transportation	0.835	0.823	0.821	0.953
Government	0.573	0.732	0.568	0.680
Environment	0.817	0.776	0.779	0.945
Social Equality	0.595	0.679	0.627	0.860
Taxation	0.762	0.779	0.741	0.914

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.