



# SocioScope - Interim Presentation

Ofir Fichman, Pavel Fadeev, Israel Peled



# Project Review

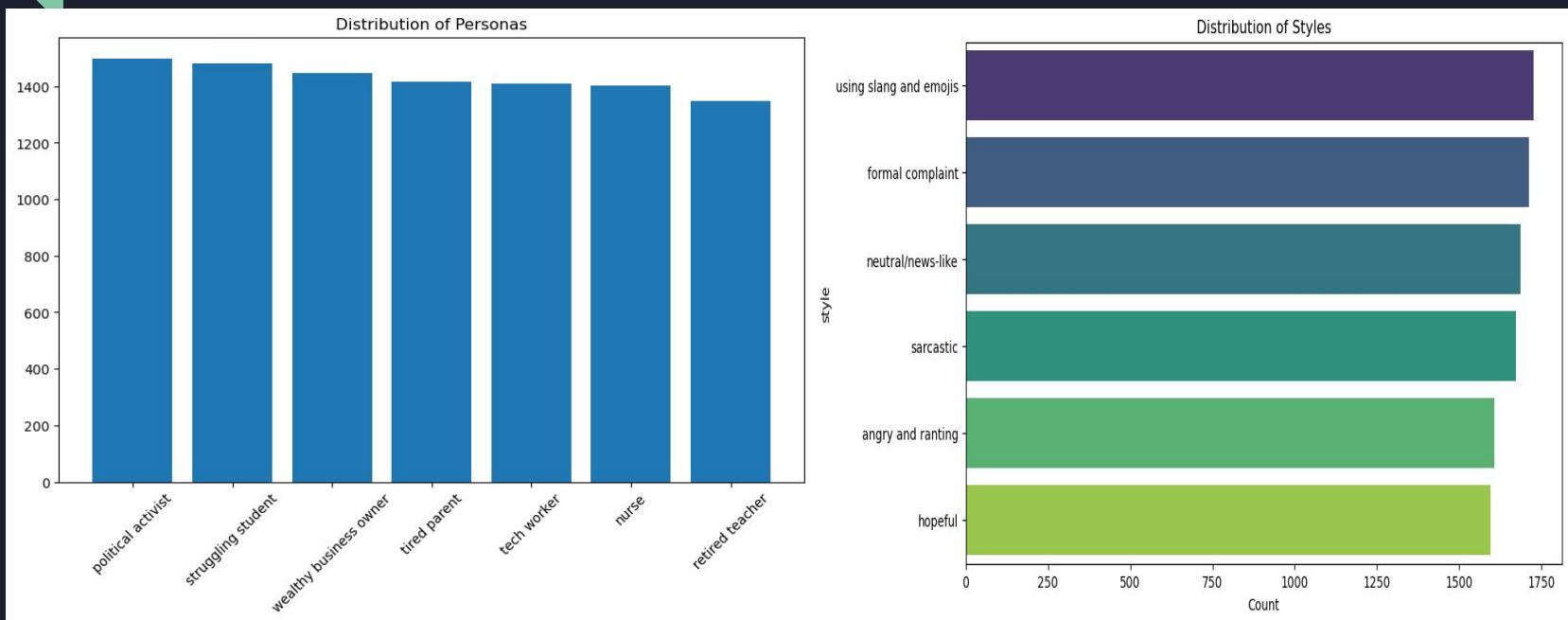
- Developing a real-time "Public Pulse" system that monitors Twitter to quantify public sentiment across 10 key socio-economic pillars.
- Transitioned from manual annotation to an automated Synthetic Data Pipeline using LLMs to generate a high-quality, 10,000-sample dataset.
- Applying advanced ABSA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

# Previous work

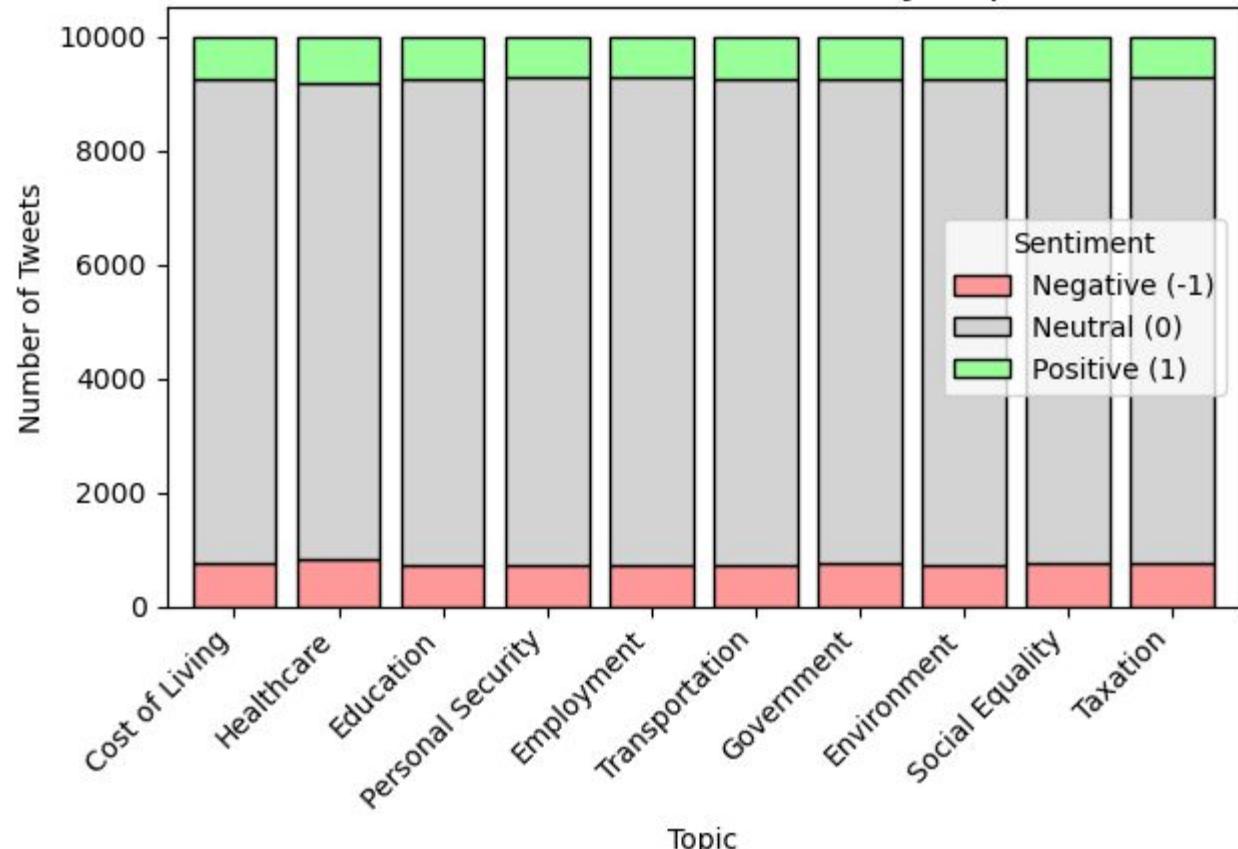
Title / Year	The Task	Methods	Data	Results	Relation to SocioScope
<a href="#"><u>MEMD-ABSA: A Multi-Element Multi-Domain Dataset for Aspect-Based Sentiment Analysis (2023)</u></a>	Extracting the Aspect, Category, Opinion, and Sentiment even when they are not explicitly mentioned.	Generative Baselines (BART/T5) + Multi-Domain Training	20,000 sentences across 5 domains (Books, Clothing, etc.)	Revealed that mining <b>implicit</b> aspects and opinions remains the biggest challenge in open-domain ABSA.	<b>Directly validates our use of LLMs</b> to infer implicit socio-economic sentiment from vague tweets.
<a href="#"><u>Aspect-Based Sentiment Analysis Using BERT(2019)</u></a>	<b>Advanced Modeling:</b> 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.
<a href="#"><u>"A novel method for ABSA: SS-LDA" (2021)</u></a>	<b>Analyzing short, sparse text:</b> Extracting topics and sentiment	SS-LDA (Statistical)	Turkish reviews + SemEval data	Proved that models can extract aspects even from very short sentences.	Directly supports the analysis of Twitter posts

# Dataset

# Distributions

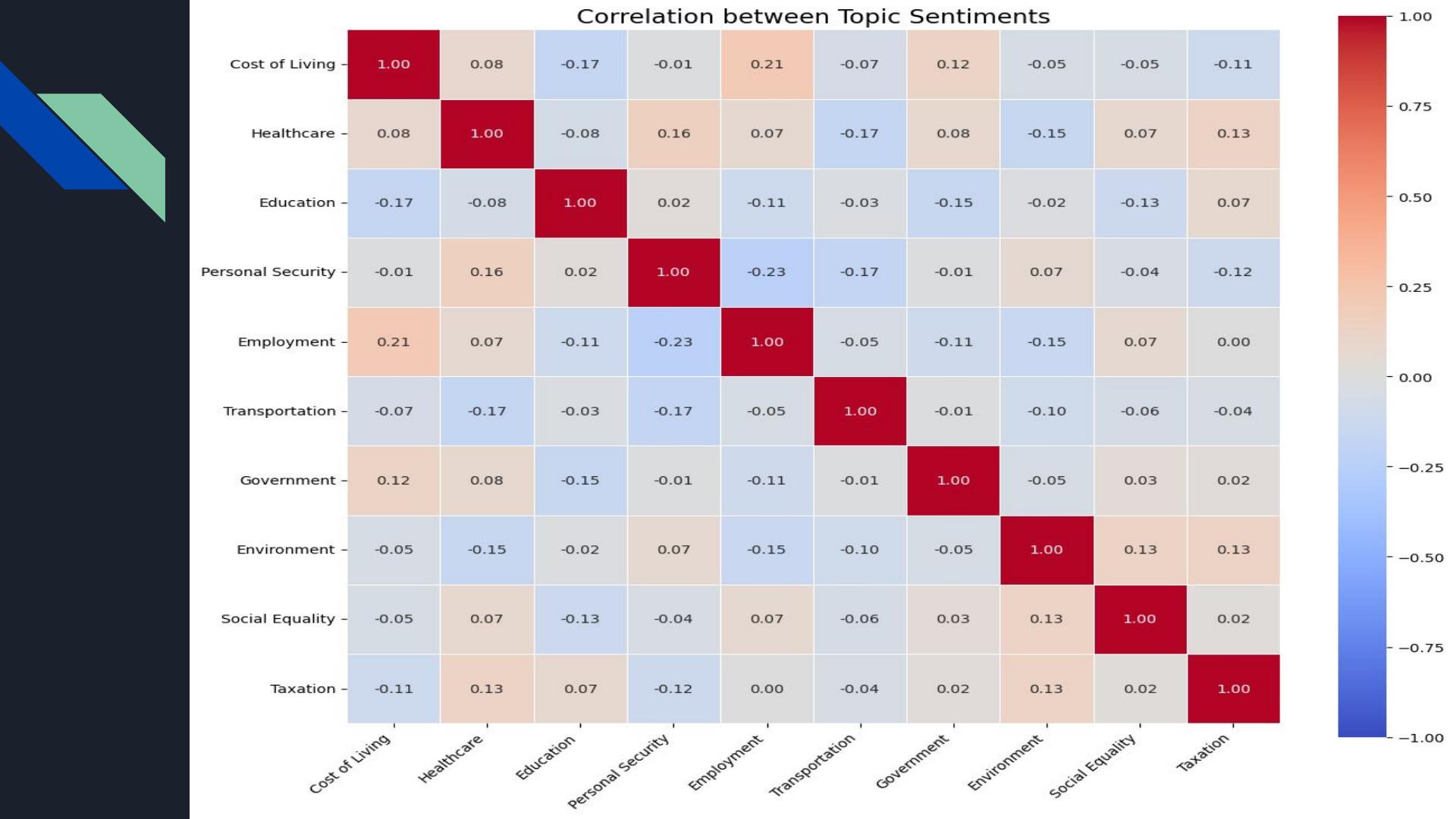


## Sentiment Distribution by Topic

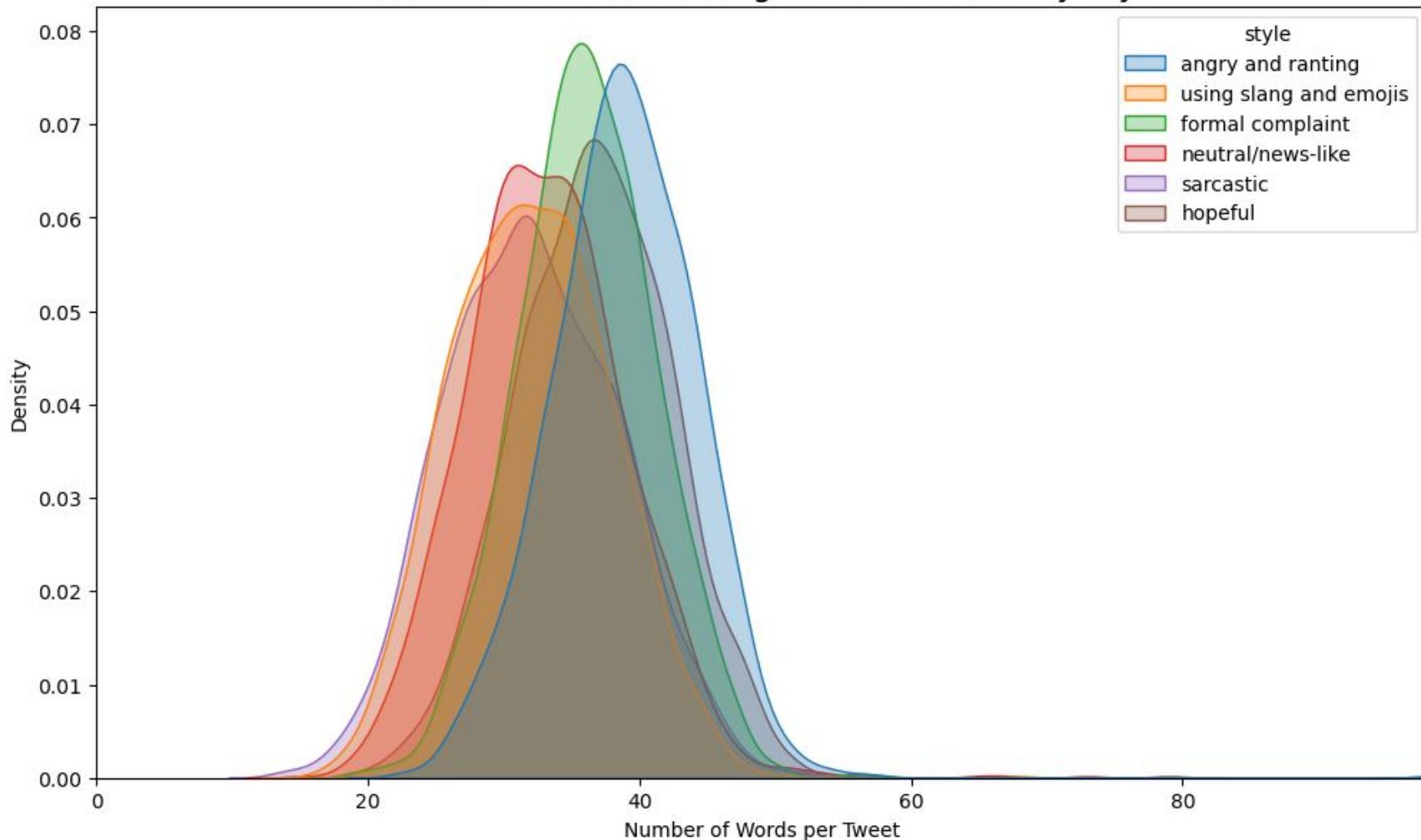


Average Sentiment (Excluding Neutrals) by Persona





# Distribution of Tweet Length (Word Count) by Style



# Baseline

```
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'][0]['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))
```

```
from tqdm import tqdm
tqdm.pandas()
sentiment_results
sampled_df['TweetText'].progress_apply(lambda x: analyze_tweet_sentiment(x))
final_df = pd.concat([sampled_df, sentiment_df], axis=1)
```

Comparison DataFrame created!  
Average Model Accuracy: 82.60%