

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light greenish-blue. They are both tilted at an angle.

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.

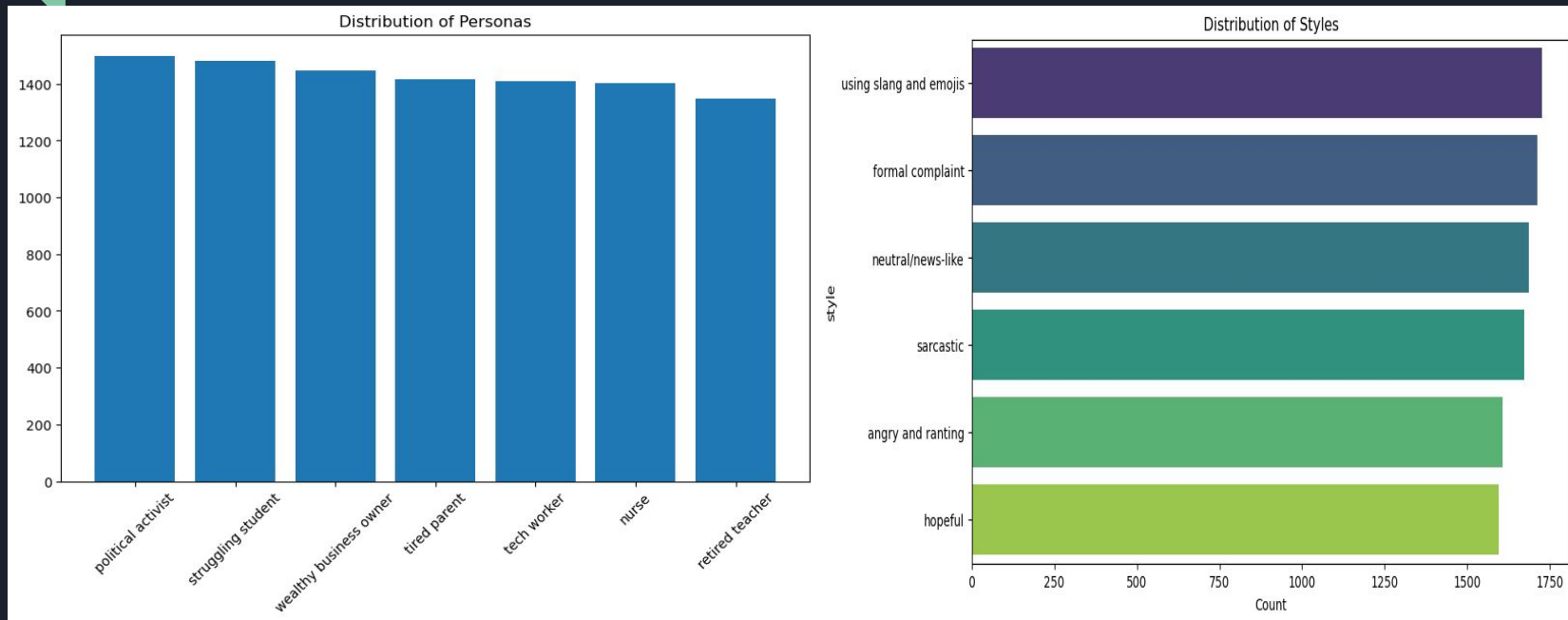
Previous work

| Title / Year | The Task | Methods | Data | Results | Relation to SocioScope |
|---|--|--|---|--|---|
| <u>MEMD-ABSA: A Multi-Element Multi-Domain Dataset for Aspect-Based Sentiment Analysis</u> (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. |
| <u>Aspect-Based Sentiment Analysis Using BERT</u> (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. |
| <u>"A novel method for ABSA: SS-LDA"</u> (2021) | Analyzing short, sparse text: 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 |

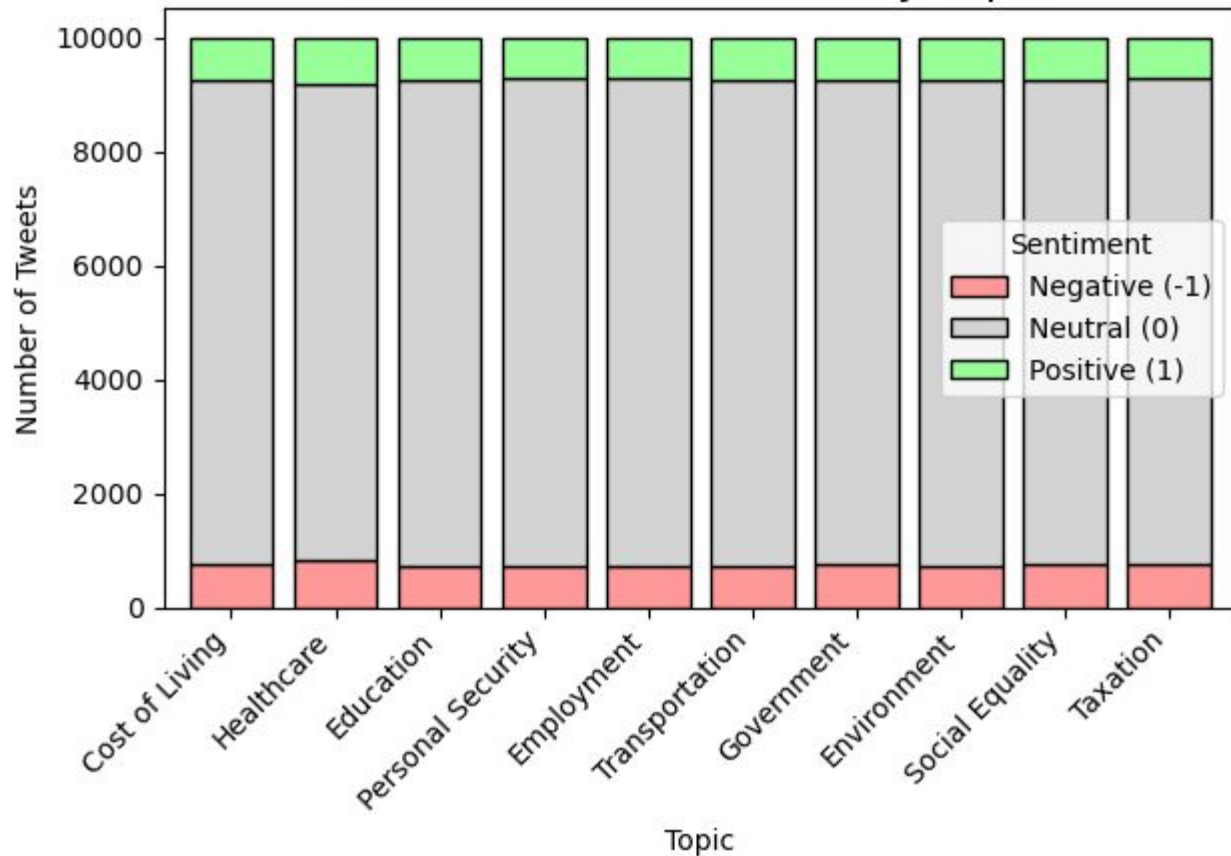
Dataset

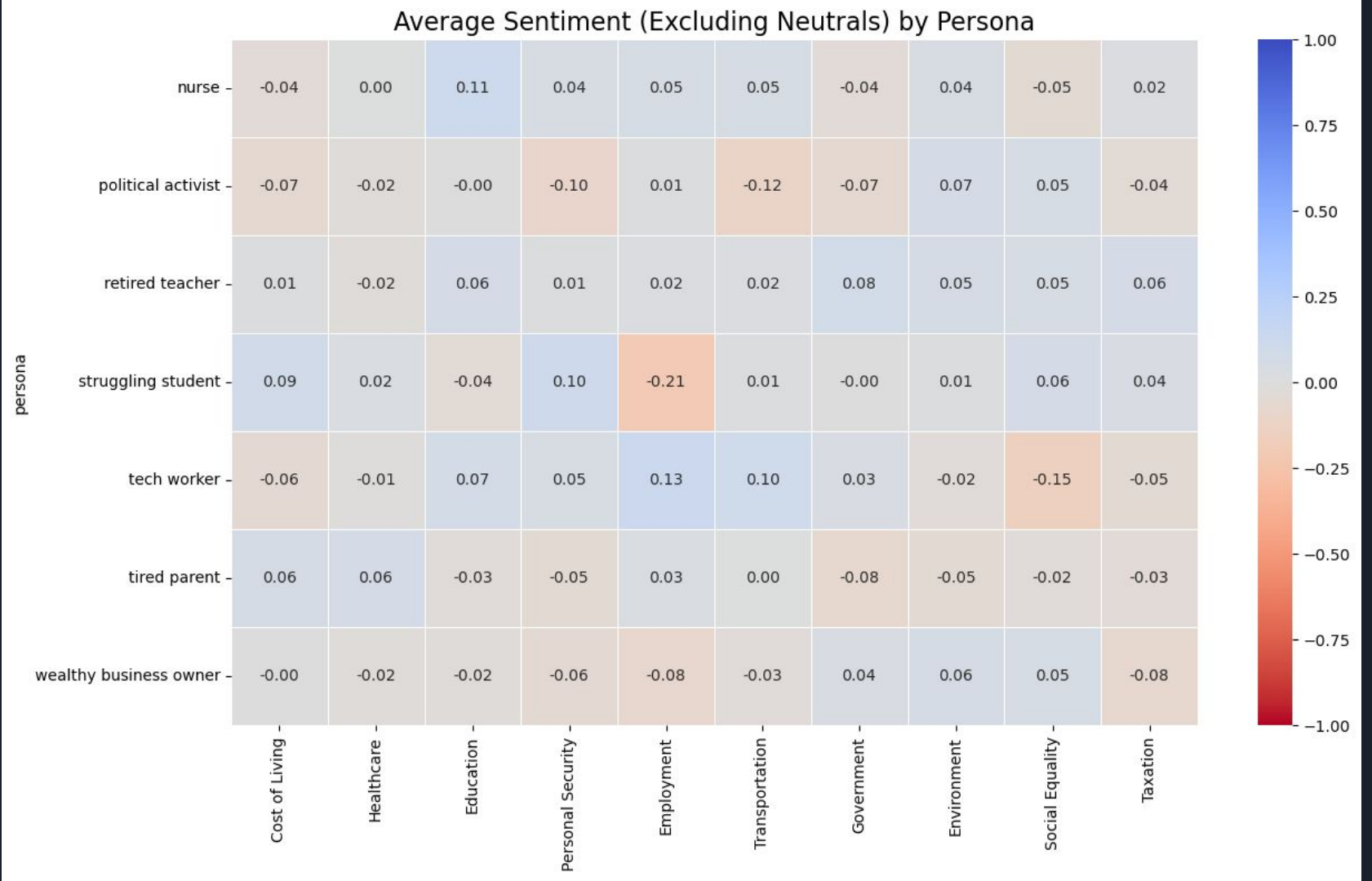
| | tweet_text | persona | style | Cost of Living | Healthcare | Education | Personal Security | Employment | Transportation | Government | Environment | Social Equality | Taxation |
|---|---|------------------------|------------------------|----------------------|------------|-----------|----------------------|------------|----------------|------------|-------------|--------------------|----------|
| 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 |

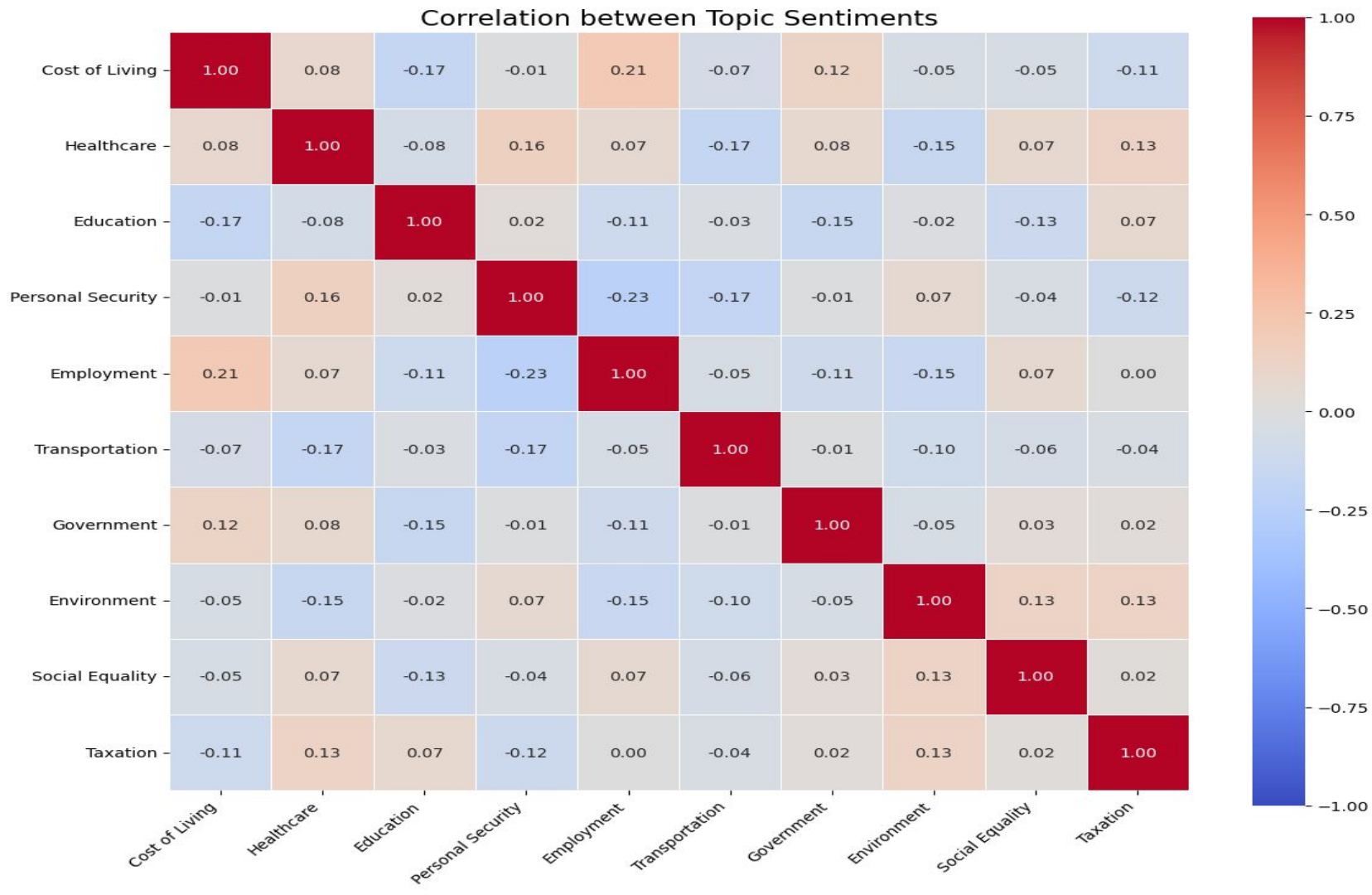
Distributions



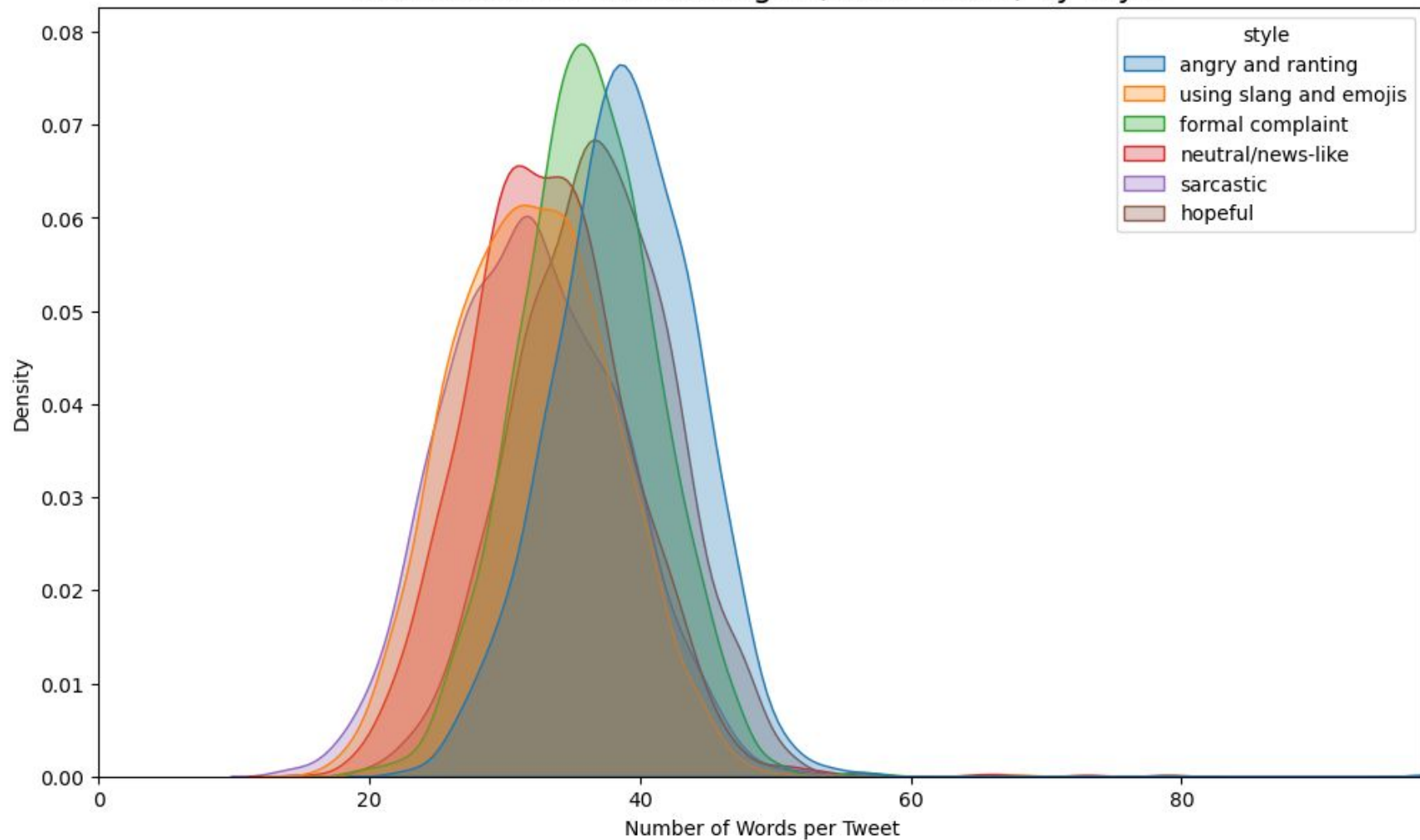
Sentiment Distribution by Topic







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']['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 = analyze_tweet_sentiment(sample_tweet)
final_df = pd.concat([sampled_df, sentiment_df], axis=1)
```

Comparison DataFrame created!

Average Model Accuracy: 82.60%