

Supplementary Materials

“The Impact of Autonomous Vehicle Accidents on Public Sentiment: A Decadal Analysis of Twitter Discourse Using roBERTa”

S1: Twitter API Queries

The Twitter API is limited to 1024 characters per request; thus, data collection was performed in two batches of keywords; the keywords were relevant to this study of messages mentioning AVs on Twitter.

Query 1: (“autonomous self-driving” OR “autonomous mobility” OR “autonomous mobility-on-demand” OR “autonomous electric vehicles” OR “autonomous electric cars” OR “autonomous electric bus” OR “autonomous electric buses” OR “autonomous electric truck” OR “autonomous electric trucks” OR “autonomous shuttle” OR “autonomous shuttles” OR “autonomous bus” OR “autonomous mini-bus” OR “autonomous minibus” OR “autonomous buses” OR “autonomous transportation” OR “autonomous vehicles” OR “autonomous vehicle” OR “autonomous cars” OR “autonomous car” OR “autonomous trucks” OR “autonomous truck” OR “autonomous driving” OR “self-driving bus” OR “self-driving mini-bus” OR “self-driving minibus” OR “self-driving buses” OR “self-driving shuttle” OR “self-driving shuttles” OR “self-driving electric vehicles” OR “self-driving electric cars” OR “self-driving taxis” OR “self-driving taxi” OR “self-driving vehicles” OR “self-driving vehicle” OR “self-driving cars” OR “self-driving car” OR “self-driving trucks”) -is:retweet lang:en

Query 2: (“self-driving truck” OR “autonomous taxi” OR “driverless bus” OR “driverless mini-bus” OR “driverless minibus” OR “driverless buses” OR “driverless shuttle” OR “driverless shuttles” OR “driverless electric vehicles” OR “driverless electric cars” OR “driverless taxis” OR “driverless taxi” OR “driverless vehicles” OR “driverless vehicle” OR “driverless cars” OR “driverless car” OR “driverless trucks” OR “driverless truck” OR “automated shuttle” OR “automated shuttles” OR “automated bus” OR “automated mini-bus” OR “automated minibus” OR “automated buses” OR “automated transportation” OR “automated vehicles” OR “automated vehicle” OR “automated cars” OR “automated car” OR “automated trucks” OR “automated truck” OR “automated driving” OR “tesla’s autopilot” OR “tesla autopilot” OR “tesla & autopilot” OR “tesla AI” OR “tesla’s AI” OR “self-driving Tesla” OR “tesla self-

driving” OR “driverless tesla” OR “autonomous tesla” OR “car autopilot” OR “car’s autopilot” OR “vehicle’s autopilot”) -is:retweet lang:en

S2: Emotion and Valence Dictionary

The Loughran-McDonald Master Dictionary¹ served as the foundation of the analysis and was augmented by relevant words identified through a text analysis conducted using QDA Miner software. Valence words such as "death" and "crash" were incorporated into the final emotion and valence dictionary.

The emotion and valence AV dictionary contained 6,281 words.

Bibliography

1. Loughran-McDonald Master Dictionary w/ Sentiment Word Lists. *University of Notre Dame*
<https://sraf.nd.edu/loughranmcdonald-master-dictionary/> (2022).

S3: Sample of 1,196 Manually Labeled Tweets Sorted By Label Type

Classification	Labels	Number of tweets	%
Multiclass classification	0 - Negative	402	33.61
	1 - Neutral	411	34.36
	2 - Positive	383	32.02

S4: Relevant Word Dictionary

QDA Miner software was used to analyze the tweet text content and identify various spellings of relevant words. Since lemmatizers such as Spacy or Stanza may not provide accurate results, a relevant dictionary was constructed to consolidate the variant spellings of a word before lemmatization. This dictionary also facilitated the selection of noteworthy words for analysis. Moreover, underscores were inserted in place of spaces between groups of relevant words to facilitate text analysis processes such as bag of words extraction and knowledge graph

construction. For example, “artificial intelligence” was transformed into “artificial_intelligence” in the dataset.

The dictionary contained 189,801 words.

S5: Binary Segmentation Methodology

1. Overview of Binary Segmentation Algorithm:

The binary segmentation algorithm [1] is a multiple change point search method [2] that serves as an approximation with “an $O(n \log n)$ computational cost, where n is the number of data points” [3]. This algorithm operates by recursively partitioning the data into segments and identifying points where significant changes in the statistical properties of the data occur [2].

2. Daily Average Sentiment Calculation:

The daily average sentiment polarity scores are computed using the mean of the sentiment values derived from individual tweets. Specifically, we calculate the average sentiment polarity as follows:

$$\text{Daily Average Sentiment} = \frac{\sum_{i=1}^n \text{Sentiment}_i}{n}$$

where n represents the total number of tweets for a given day, and where Sentiment_i represents the sentiment polarity score assigned to each tweet. In our analysis, these scores can take discrete values of -1 (negative), 0 (neutral), or 1 (positive), as well as continuous values derived from a sentiment analysis model.

To handle multiple tweets per day, we utilize the ‘resample’ function in Pandas to group tweets by day on the basis of their timestamp.

3. Specific Parameters and Criteria for Period Identification:

In our implementation, we utilized the following parameters:

a) Model Specification:

We employed an l2-norm penalty model (model = "l2"), which enhances prediction reliability by reducing the magnitude of the coefficients representing each factor's influence on the outcome [4]. This approach penalizes larger

coefficients, thereby stabilizing the model and mitigating issues that may arise when factors are closely related. Consequently, predictions become more robust and less sensitive to minor data fluctuations. In the context of change point detection, employing an l_2 -norm penalty further stabilizes the process by discouraging overly complex or noisy segment partitions, thus improving the algorithm's capacity to identify significant change points.

b) Number of Change Points:

We specified `n_bkps=1`, indicating that we sought to identify one change point within our time series data (resulting in two sentiment periods). Additionally, we set `n_bkps=3` to detect three change points (yielding four sentiment periods). This methodological approach enables the identification of multiple significant shifts in sentiment over time, thereby providing a more nuanced understanding of the evolution of public opinion. The selection of these parameters allows for a comprehensive analysis of temporal trends in sentiment, capturing both major and subtle changes in the public perception of autonomous vehicle technology.

4. Technical Details of Implementation:

The implementation was conducted using the Ruptures library in Python, which is specifically designed for change point detection in time series data. The relevant code snippet is provided below:

```
Python code:
import ruptures as rpt

# Define the model
model = "l2" # l2-norm penalty
algo = rpt.Binseg(model=model).fit(sentiment_values)

# Predict change points
result = algo.predict(n_bkps=3) # for 4 sentiment period definition
```

```
# Print detected change points  
change_points = [daily_sentiment.index[cp] for cp in result if cp <  
len(daily_sentiment.index)]  
print("Detected change points:", change_points)
```

In this code, `sentiment\_values` is a NumPy array containing the previously computed daily average sentiment scores. The `fit` method applies the binary segmentation algorithm to these data, whereas `predict` returns the identified change points.

5. Statistical Justification for Identified Change Points:

The statistical justification for identifying change points lies in evaluating how well each segment fits a specified model compared with when a change point is introduced. The l2-norm penalty helps ensure that segments are as statistically distinct as possible from one another by minimizing the variance within segments while maximizing the variance between them.

6. Validation Methods for Period Boundaries:

To validate the identified change points, we visually inspected the results by plotting the daily sentiment scores and the detected change points on a graph, alongside the global events potentially connected with the change points. This approach allows for an assessment of whether shifts in sentiment correspond with known events or changes in public opinion. Furthermore, we conducted sensitivity analyses by varying parameters (e.g., changing `n\_bkps`) to determine if similar change points are consistently identified across different configurations.

S6: Sentiment Score and Number of Tweets for the Two Defined Periods

The sentiment polarity analysis over a decade indicated two distinct periods, with a pivotal turning point on June 23, 2016. Notably, period 1 spanned from January 1, 2012, to June 22, 2016, and the ensuing period extended from June 23, 2016, to December 31, 2021. The subperiods were defined using a binary segmentation algorithm, as follows:

- Period 1A (January 1, 2012, to April 29, 2014);
- Period 1B (April 30, 2014, to June 22, 2016);
- Period 2A (June 23, 2016, to March 14, 2018);
- Period 2B (March 15, 2018, to December 31, 2021).

Periods	Number of tweets	% of tweets	Average sentiment score	Standard deviation
Period_1	114,14	22.47	0.046	0.822
Period_1A	25,742	5.07	0.124	0.808
Period_1B	88,398	17.41	0.024	0.825
Period_2	393,733	77.53	-0.121	0.857
Period_2A	119,151	23.46	-0.062	0.837
Period_2B	274,582	54.07	-0.147	0.865

S7: Term Frequency-Inverse Document Frequency (TF-IDF) Scores and Polarity Sentiment Scores for the Top 20 Words in the Two Defined Periods

The textual data were divided into two periods as identified by the binary segmentation method (period 1: 2012-2016; period 2: 2016-2021). Term frequency-inverse document frequency (TF-IDF) scores were then computed, and the top 20 words were selected for each period. Each of these words was subsequently searched within individual tweets to collect the associated sentiment, and the mean sentiment score (-1 for negative, 0 for neutral, and +1 for positive) was calculated for each word.

- Data segmentation:

We define the periods as follows:

$$T = \{T_1, T_2\}$$

where:

$$T_1 = \{t \mid 2012 - 01 - 01 \leq t \leq 2016 - 06 - 22\},$$

$$T_2 = \{t \mid 2016 - 06 - 23 \leq t \leq 2021 - 12 - 31\}$$

- TF-IDF Calculation:

The TF-IDF for a term t in document d , relative to the entire document set D , is given by:

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

where:

$$TF(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

is the term frequency, with $f_{t,d}$ being the frequency of term t in document d .

$$IDF(t, D) = \log\left(\frac{|D|}{|\{d \in D: t \in d\}|}\right)$$

is the inverse document frequency, where $|D|$ is the total number of documents, and where $|\{d \in D: t \in d\}|$ is the number of documents containing term t .

- Top Word Selection:

For each period T_i (where $i \in \{1,2\}$), the top 20 words are selected on the basis of their TF-IDF scores. The set of top words for period T_i is:

$$W_i = \{w_1, w_2, \dots, w_{20}\} \text{ such that } \max(\text{TF-IDF}(w, d, D) \text{ for } i \in \{1,2\})$$

- Polarity Sentiment Calculation:

For each word $w \in W_i$, the sentiment scores associated with its occurrence in the tweets are collected as follows:

$$S_{(w)} = \{s_j \mid s_j \in \{-1, 0, 1\}, j = 1, \dots, n\}$$

where s_j is the sentiment score of the j -th occurrence of word w , and where n is the total number of occurrences of w .

The mean sentiment score (MSS) for each word w is then calculated as:

$$MSS_{(w)} = \frac{1}{n} \sum_{j=1}^n s_j$$

- Final output:

For each period T_i , the output is the set of words W_i along with their mean sentiment scores $MSS_{(w)}$:

$$\forall T_i, W_i: \{(w, MSS_w \mid w \in W_i)\}$$

This captures the process from data segmentation to sentiment analysis in a concise mathematical form.

Period 1 results

Word	TF-IDF Score (Period_1)	Polarity Sentiment Score (Period_1)
google	5247.897	0.023
road	1716.315	0.088
technology	1588.398	0.070
future	1499.502	0.242

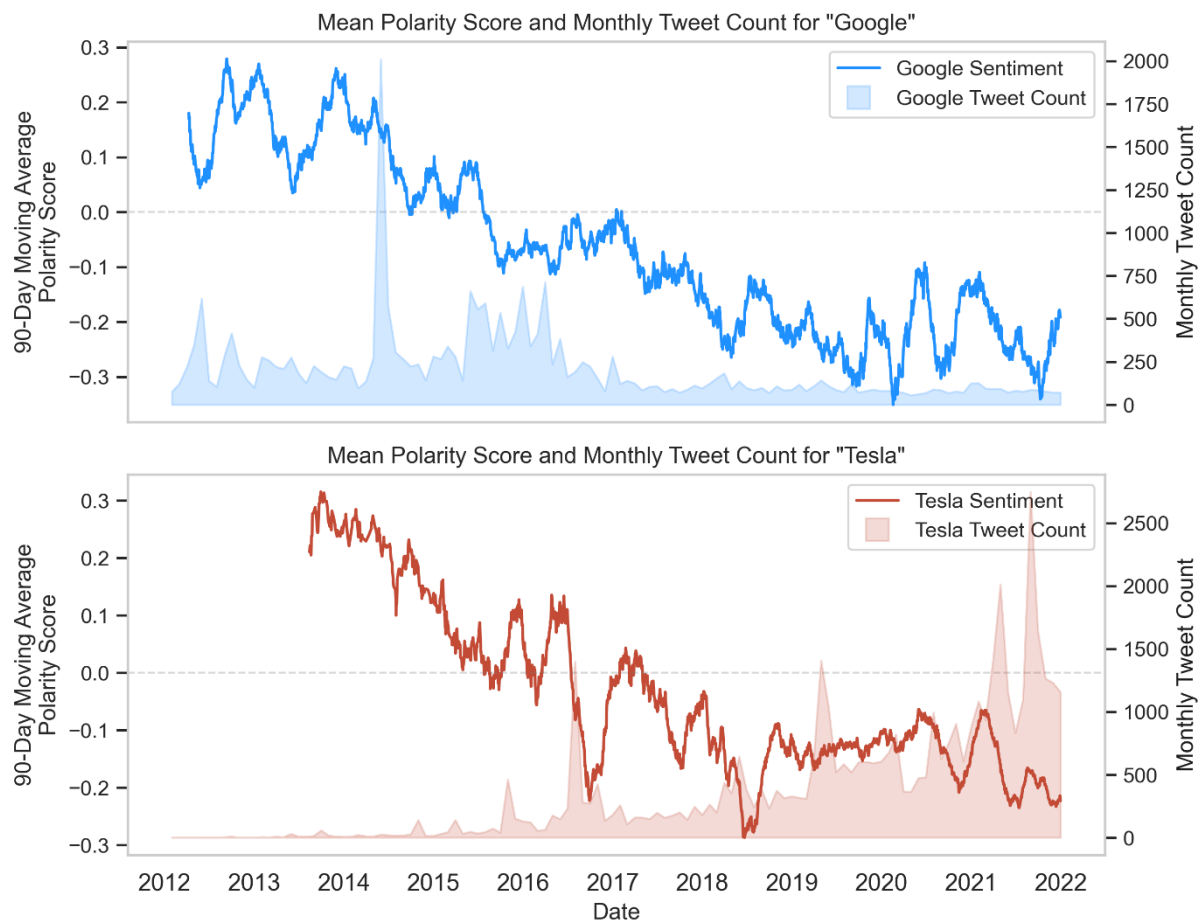
Word	TF-IDF Score (Period_1)	Polarity Sentiment Score (Period_1)
accident	1337.373	0.094
great	1024.139	0.631
wait	1004.881	0.514
human	915.830	0.025
safe	912.223	0.362
love	839.807	0.636
cool	838.102	0.657
uber	791.707	-0.027
hope	788.474	0.421
problem	747.962	-0.107
company	721.674	-0.045
stop	720.164	-0.010
traffic	707.120	0.294
today	683.364	0.086
life	667.252	0.354
interesting	667.136	0.278

Period 2 results

Word	TF-IDF Score (Period_2)	Polarity Sentiment Score (Period_2)
tesla	10747.224	-0.178
technology	10073.787	-0.093
road	8998.873	-0.063
artificial intelligence	7169.939	-0.131
future	7042.799	0.098
human	6987.915	-0.095
company	6757.292	-0.199
uber	6594.624	-0.308
tesla autopilot	6198.219	-0.207
safe	5729.510	0.150
kill	5527.672	-0.570
accident	5086.674	-0.028
stop	5002.889	-0.184
problem	5000.591	-0.270
google	4669.940	-0.183
wait	4495.174	0.223
great	4423.790	0.480
life	4125.239	0.132
traffic	3900.676	0.144
robot	3844.598	-0.169

S8: Trends in Sentiment and Tweet Volume for Mentions of Google and Tesla

The 90-day moving average of tweets mentioning Tesla and Google indicates that the evolution of sentiment polarity over the studied decade has shifted from positive to negative. The plot further reveals that Google received significantly more mentions during the initial positive period (2012-2016), whereas Tesla was predominantly cited during the subsequent negative period. Notably, the volume of tweets mentioning Tesla is approximately twice that of Google, with totals of 43,777 and 24,257 tweets, respectively.



References

1. Scott AJ, Knott M. A Cluster Analysis Method for Grouping Means in the Analysis of Variance. *Biometrics* (1974) **30**:507–512. doi:10.2307/2529204
2. Fryzlewicz P. Wild Binary Segmentation for Multiple Change-Point Detection. *The Annals of Statistics* (2014) **42**:2243–2281.
3. Killick R, Fearnhead P, Eckley IA. Optimal Detection of Changepoints With a Linear Computational Cost. *Journal of the American Statistical Association* (2012) **107**:1590–1598. doi:10.1080/01621459.2012.737745
4. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. New York, NY: Springer New York (2009). doi:10.1007/978-0-387-84858-7