

Chapter3_LI HONGLIN.pdf

by HONGLIN LI

Submission date: 27-Jun-2025 07:00AM (UTC-0700)

Submission ID: 2696249726

File name: Chapter3_LI_HONGLIN.pdf (755.98K)

Word count: 1126

Character count: 6046

RESEARCH METHODOLOGY

3.1 Introduction

This chapter is about how to analyze what people feel towards Trump's China 2025 tariff policy on "X". It discusses the entire process of data acquisition and cleaning to determine the feelings using the VADER analysis tool. There are three central concerns this study will focus on: people's expectation of the policy's effects, policy announcements, and policy updates. It will provide real evidence of a shift in public sentiments.

3.2 Research Framework

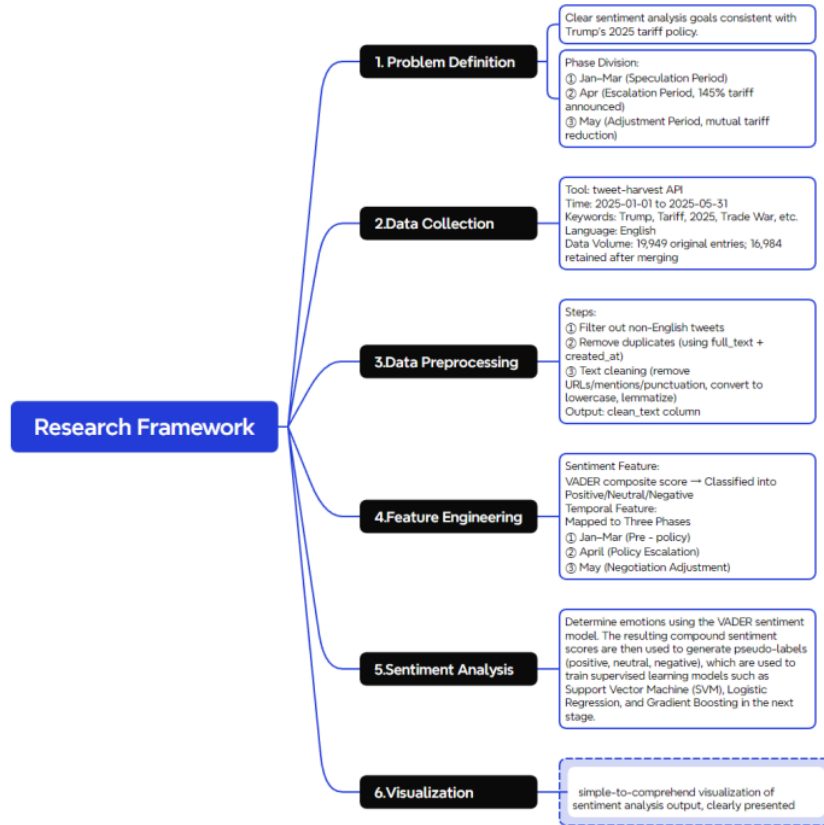
The research framework follows a standard data science project life cycle and is divided into the following stages:

1. Question definition: Clear sentiment analysis goals consistent with Trump's 2025 tariff policy.
2. Data collection: Collect the significant "X" data with the appropriate keywords and in a particular time span.
3. Data preparation: Prepare the data and clean it thoroughly in order to make it more credible.

4. Feature construction: Develop features from emotions and time in a correct analysis.

5. Sentiment analysis: Determine emotions using VADER emotion model. The resulting sentiment scores are then used to generate pseudo-labels to train a supervised learning model in the next stage.

6. Visualization; simple-to-comprehend visualization of sentiment analysis output, clearly presented



3.1 Research Framework

3.3 Problem Formulation

This research is interested in examining how individuals' perspectives are altered at three phases of policy.

January-March 2025: Retrospection on policies and initial responses

New policy adjustments and 145% tariffs in April 2025 were announced.

May 2025: Policy changes and reduced tariffs among nations.

Key objectives:

Apply VADER to determine whether tweets are positive, neutral, or negative.

Examine how individuals' attitudes shifted during three different time frames.

3.4 Data Sources & Collection



Combine the data mined separately every month into a data set."

```
[1] # prompt: 加载云存储盘
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

import pandas as pd
import os

# Set your folder path
folder_path = '/content/drive/MyDrive/Colab Notebooks/Untitled Folder'

# Initialize an empty DataFrame
combined_data = pd.DataFrame()

# Iterate through the folder and merge all Excel files
for file_name in os.listdir(folder_path):
    if file_name.endswith('.csv'): # Check if the file is an Excel file
        file_path = os.path.join(folder_path, file_name)
        try:
            # Read Excel file
            data = pd.read_csv(file_path)
        except Exception as e:
            print(f'Error reading {file_name}: {e}')
            continue # Skip the file if there's an error

        # Combine data
        combined_data = pd.concat([combined_data, data], ignore_index=True)

# Save the merged file as a CSV
output_path = '/content/drive/MyDrive/Colab Notebooks/AllTwitter.csv'
combined_data.to_csv(output_path, index=False)

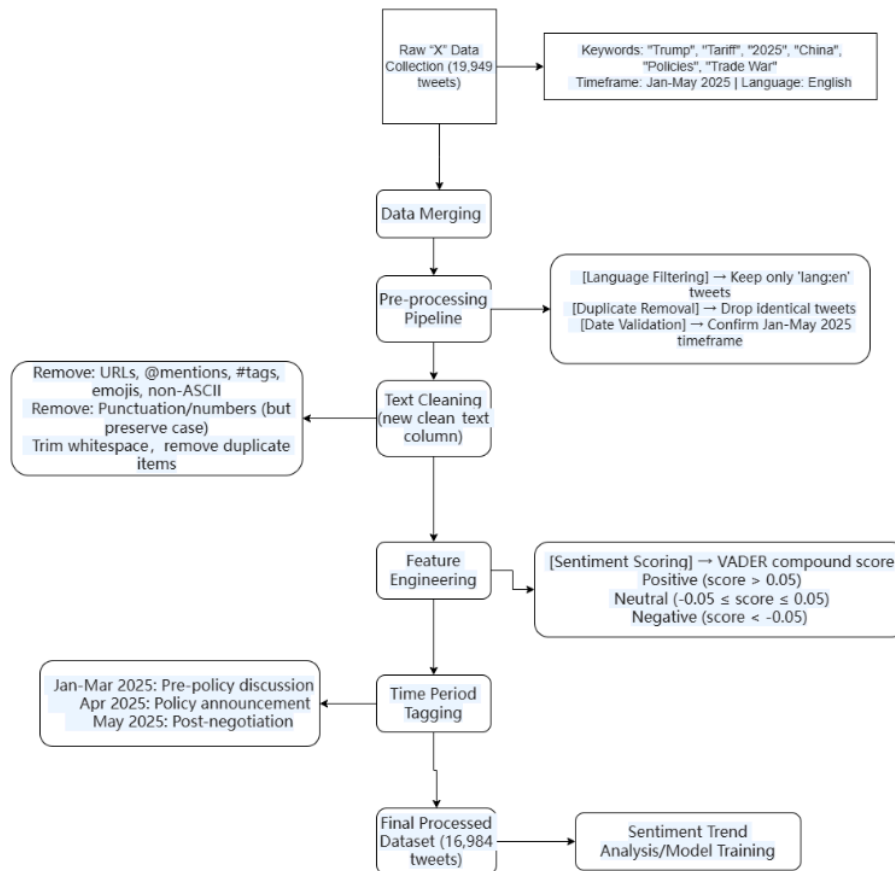
print(f'File merged successfully! Merged file saved at: {output_path}')
```

3.3 Data Merging

The total data set collected is 19,949 rows of data, including 15 columns.

AllTwitter.csv														
Open with														
date_of_tweet	tweet_text	tweet_id	tweet_time	tweet_location	tweet_language	tweet_reply_count	tweet_retweet_count	tweet_like_count	tweet_hashtag_count	tweet_hashtag_text	tweet_hashtag_id	tweet_hashtag_name	tweet_hashtag_image	tweet_hashtag_video
2020-01-01 00:00:00	1	1000000000000000000	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	2	1000000000000000001	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	3	1000000000000000002	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	4	1000000000000000003	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	5	1000000000000000004	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	6	1000000000000000005	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	7	1000000000000000006	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	8	1000000000000000007	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	9	1000000000000000008	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	10	1000000000000000009	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	11	1000000000000000010	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	12	1000000000000000011	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	13	1000000000000000012	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	14	1000000000000000013	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	15	1000000000000000014	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	16	1000000000000000015	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	17	1000000000000000016	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	18	1000000000000000017	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	19	1000000000000000018	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	20	1000000000000000019	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	21	1000000000000000020	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	22	1000000000000000021	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	23	1000000000000000022	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	24	1000000000000000023	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	25	1000000000000000024	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	26	1000000000000000025	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	27	1000000000000000026	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	28	1000000000000000027	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	29	1000000000000000028	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	30	1000000000000000029	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	31	1000000000000000030	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	32	1000000000000000031	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	33	1000000000000000032	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	34	1000000000000000033	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	35	1000000000000000034	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	36	1000000000000000035	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	37	1000000000000000036	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	38	1000000000000000037	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	39	1000000000000000038	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	40	1000000000000000039	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	41	1000000000000000040	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	42	1000000000000000041	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	43	1000000000000000042	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	44	1000000000000000043	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	45	1000000000000000044	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	46	1000000000000000045	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	47	1000000000000000046	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	48	1000000000000000047	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	49	1000000000000000048	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	50	1000000000000000049	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	51	1000000000000000050	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	52	1000000000000000051	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	53	1000000000000000052	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	54	1000000000000000053	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	55	1000000000000000054	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	56	1000000000000000055	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	57	1000000000000000056	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	58	1000000000000000057	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	59	1000000000000000058	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	60	1000000000000000059	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	61	1000000000000000060	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	62	1000000000000000061	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	63	1000000000000000062	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	64	1000000000000000063	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	65	1000000000000000064	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	66	1000000000000000065	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	67	1000000000000000066	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	68	1000000000000000067	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	69	1000000000000000068	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	70	1000000000000000069	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	71	1000000000000000070	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	72	1000000000000000071	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	73	1000000000000000072	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	74	1000000000000000073	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	75	1000000000000000074	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	76	1000000000000000075	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	77	1000000000000000076	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	78	1000000000000000077	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	79	1000000000000000078	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	80	1000000000000000079	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	81	1000000000000000080	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	82	1000000000000000081	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	83	1000000000000000082	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	84	1000000000000000083	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	85	1000000000000000084	2020-01-01 00:00:00		en	0	0	0	0					
2020-01-01 00:00:00	86	1000000000000000085	20											

3.5 Data Pre-processing



Once we filtered the languages, deleted duplicates, and eliminated noise, we had 16,984 tweets. This is our final data set that we will examine in terms of sentiment scores and trends through time.

```
print(df[["created_at", "full_text", "clean_text"]].head())
```

	created_at	full_text	clean_text
0	Thu Jan 30 23:59:19 +0000 2025	@unusual_whales Threat? He literally did this ...	threat he literally did this in people wake u...
1	Thu Jan 30 23:58:26 +0000 2025	Trump's 25% tariff on Canada and Mexico; a hig...	trumps tariff on canada and mexico a higher o...
2	Thu Jan 30 23:48:51 +0000 2025	Trump imposes 25% tariffs on Canada and Mexico...	trump imposes tariffs on canada and mexico fr...
3	Thu Jan 30 23:48:41 +0000 2025	Trump imposes 25% tariffs on Canada and Mexico...	trump imposes tariffs on canada and mexico fr...
4	Thu Jan 30 23:48:25 +0000 2025	@CanadaFreedom0 This is why Trump is tariff fo...	this is why trump is tariff focused mexico t...

```
print("Total tweets after cleaning:", len(df))
```

Total tweets after cleaning: 16984

3.5 Processed data

There are two key steps in data preprocessing. These steps ensure the data is clean, consistent, and prepared for sentiment analysis using the VADER model.

3.5.1 Preliminary analysis

- Verify that the data is correct for the target period (January to May 2025).
- There is a simple check to determine the number of tweets posted during a particular period. This is to ensure the dates listed in the policy (for instance, approximately April 10, 2025) are accurate.
- Ensure the lang field exists and is solely utilized for filtering tweets that are in English.

3.5.2 Data cleaning

To ensure that the tweets are good for the VADER model, we performed some cleaning steps. We retained all the emotional parts but got rid of typical noise in social media text.

Gradually clean the logic:

Filter English tweets

Only the English notes were preserved according to Vader's list of emotional words in the English language.

Remove duplicate data

Use the entire text and place in columns to eliminate duplicate tweets so that every comment is counted once only.

```
import pandas as pd
import re

# 读取数据文件
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/ALLTwitter1.csv')
df = df[df["lang"] == "en"].copy()

# 步骤 2: 去除重复推文 (同文本同时间被视为重复)
df = df.drop_duplicates(subset=["full_text", "created_at"])

# 步骤 3: 定义清洗函数
def clean_text(text):
    text = re.sub(r"http\S+|www\S+", "", text) # 删除URL
    text = re.sub(r"@w+", "", text) # 删除提及
    text = re.sub(r"#w+", "", text) # 删除hashtag
    text = re.sub(r"[^\w\s]", "", text) # 删除标点符号
    text = re.sub(r"\d+", "", text) # 删除数字
    text = text.lower().strip() # 小写化并去空格
    return text

# 步骤 4: 应用清洗函数
df["clean_text"] = df["full_text"].astype(str).apply(clean_text)

# 可选: 显示前几行结果检查
print(df[["created_at", "full_text", "clean_text"]].head())
```

```
created_at \
0 Thu Jan 30 23:59:19 +0000 2025
1 Thu Jan 30 23:58:26 +0000 2025
2 Thu Jan 30 23:48:51 +0000 2025
3 Thu Jan 30 23:48:41 +0000 2025
4 Thu Jan 30 23:48:25 +0000 2025

full_text \
0 @unusual_whales Threat? He literally did this ...
1 Trump's 25% tariff on Canada and Mexico; a big...
2 Trump imposes 25% tariffs on Canada and Mexico...
3 Trump imposes 25% tariffs on Canada and Mexico...
4 @CanadaFreedom0 This is why Trump is tariff fo...
```

3.6 Data cleaning

Text preprocessing

Cleaning text

1. Delete links: VADER does not comprehend links; they will be puzzling.
2. Remove mention (@user): mention feels emotional.
3. Take away tag: Tag sign is removed but the word can be retained to facilitate the understanding.
4. Eliminate emoji and non-ASCII characters: These are excluded since they are not text input.
5. Remove punctuation and numbers: Vader explains words, not symbols or numbers.
6. no conversion to lowercase: standardized text. e.g., "tariffs" is equivalent to "tariffs".
7. Trim blank areas: Remove excess space for consistency.

8. Clean storage results:

9. The cleaned result is saved in a new column called `clean_text`, which is used as input for sentiment analysis

3.6 Feature Engineering

Improve tweet data to analyze emotions and examine over time trends. Improve the meaning and time components. VADER is frequently employed since it is suitable for social media text (Chavan et al., 2024;) (Gandy et al., 2025), and parts of it were employed in sorting emotions.

In order to further improve the effect of sentiment classification, this study not only uses the Compound scores generated by the VADER model for trend analysis, but also uses them as pseudo-labels for model training. A balanced labeled dataset is constructed from these pseudo-labels, which will be used to train supervised learning models such as support vector machines, random forests, and logistic regression. This semi-supervised modeling strategy will be elaborated in the next chapter.

3.6.1 Emotional Feature Extraction

The cleaned tweet (`clean_text`) is then analyzed using the VADER sentiment analyzer in order to obtain a score representing the overall mood of a tweet.

Emotional tags are assigned according to these levels of scores:

- Positive:Compound score > 0.05
- Neutral:-0.05 ≤Compound score ≤ 0.05
- Negative: compound score <-0.05

3.6.2 Time feature generation

Each tweet is tagged with a particular date and one of three various policy times:

- January to March 2025: Discuss it and consider thoroughly before legislating.
- April 2025: Peak tariff policy announcement and reactions of other nations.
- May 2025: Post-adjustment period after reciprocity negotiations

3.8 Results visualization

Displaying results The labeled emotional data is plotted on a chart in order to observe public opinion trends over time.

Line graph: It demonstrates how the average rating varies with different months, in relation to significant policy events.

Vertical bar chart: It displays the number of positive, negative, and neutral tweets during each phase of the policy.

Word cloud: Display typical words in various emotional groups so that people can comprehend what the public is interested in.

Python tools such as matplotlib, seaborn, and wordcloud are employed in visualization to represent emotional trends simply and simply.

These tweets are classified based on VADER's (valence-aware dictionary and emotion inference) compound score. This is suitable to classify the brief and informal messages in Twitter tweets.

3.9 Summary

This chapter shows how to set up an analysis process based on the VADER model. In the next chapter, we will further explain not only the three-stage sentiment trend, but also how VADER scores are used to generate pseudo-labels and then build a semi-supervised sentiment classification model.

Chapter3_LI HONGLIN.pdf

ORIGINALITY REPORT

4%

SIMILARITY INDEX

4%

INTERNET SOURCES

2%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

www.readkong.com

Internet Source

1%

2

pmc.ncbi.nlm.nih.gov

Internet Source

1%

3

portfolios.cs.earlham.edu

Internet Source

1%

4

www.coursehero.com

Internet Source

1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On