

POP77142 - Quantitative Text Analysis Assignment

100 years evolution of American values through presidential inauguration speeches

Abstract

This paper analysed 26 American presidential inauguration speeches to identify Schwartz's human values trend in the last 100 years. The dictionary method and LLM method using Mistral LLM were adopted and compared in detail with several case studies. The result shows a slight advantage in favor of the LLM model with its ability to comprehend subtle wording that would be impossible for a dictionary to detect. Based on the result of the LLM method, self-transcendence was the defining value of the Democrats with some openness mixing in. Conservation value has been declining since 1925 and is nowhere to be found recently. For Republicans, the value landscape is a tug of war between self-transcendence and conservation with conservation on the rise recently. Self-enhancement values are rare to find in both parties over the last 100 years but has found its resurgence in the recent two Trump's administrations.

Introduction

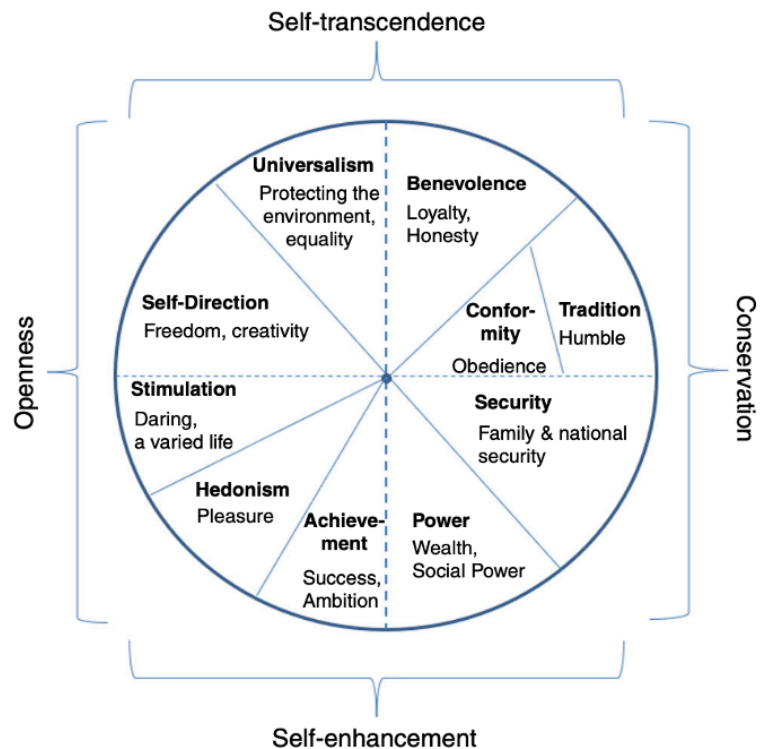
In the last several years, we have seen a number of debates about whether the US incumbent Democrat administration has become too left-wing in the media. In the last few months, within the start of the new presidential term, we now hear the debate about the current administration has become too right-wing. In the heat of polarization, if we want to find a sanctuary of common ground for the American people, I think one of the best places to look at is the presidential inauguration speech. After fierce partisan campaign competition, the speech served as a call for unity and reassessment of the common American values of the past, present and future. In this research paper, I will discover how the American presidential inauguration speech's portraits of American values have evolved in the last 100 years. I will use two methods side by side: a classic text as data approach - the dictionary approach and a more modern large language mode (LLM) approach to help uncover the speeches' value sentiment.

Literature review

The content of presidential speeches is a common subject of dissection. Stuckey (2010) describes inaugural addresses as providing a national reconciliation function. It attempts to do so through the use of affirmative rhetoric. Through this the president calls "upon shared national values and ideals and enunciates broad national political principles.". There is a wide body of literature analyzing the inaugural speech using text-as-data technique. Kubát et al.(2016) focused on using moving average type-token ratio (MATTR) and secondary thematic concentration (STC) to analyse the richness of vocabulary and the level of focus on any topics of the speeches. Light (2014) used the method of network text analysis to find communities of themes and similarity of different presidential addresses. Analyzing values within the presidential inauguration speech are usually done in a qualitative method. They usually focused on a single speeches or a handful of speeches e.g. Zhou (2023)'s analysis of Trump or West (2014)'s analysis of Obama. Analysing presidential inauguration speech quantitatively using Schwartz's value framework allows us to rigorously evaluate the trend of these values's manifestation in the last 100 years in America.

The most widely used set of human values has been developed by Schwartz (1992). They are usually estimated by a series of questionnaires that measure 10 lower order values that are also sorted into 4 higher order values. Figure 1 showed how the values are structured in a circular way.

Figure 1: 10 types of personal values by Schwartz



The closer the values in the circle, the more likely they are inhibited in the same individual. In Europe, the 21-item questionnaire is included every year since round 1 of the ESS. This provides a convenient way to gauge the trajectory of European human values. There is no such longitudinal project in the US related to Schwartz's human values. This paper can serve as a glimpse into how the American values have evolved in the last 100 years through the framework of Schwartz. Though my impression is that the speeches more likely reflected the values that American want to strive for at

that specific moment in history, rather than the values its people actually held. My hypothesis is that the value of self-transcendence has been rising over the century due to the increasingly larger audience of the speech, making the pressure to show a message of unity, caring and inclusion. I also predict a rise in openness in the mid 90s and then a decline due to the rise of anti immigration sentiment in recent decades. I also expect the LLM approach to be more accurate in capturing the general tone of the speech compared to the dictionary based approach.

Data and Methodology

Presidential inauguration speech was scrapped from the American Presidency Project¹. Beautiful Soup library in Python was used to extract the data from the html file. Starting from President Calvin Coolidge in 1925 to President Donald Trump in 2025, there are a total of 26 presidential terms. 26 Inauguration speeches were successfully collected from the website. For the creation of the dictionary, the following prompt was used to get an initial set of 50 words for each value: "Create 4 python lists each with 50 words that represent the 4 higher Schwartz values which will be used to identify values of the US presidential inauguration speech ". After examining each dictionary, I have found the lists are a good representation of what the values are with just some minor removal of inappropriate words. I also inserted words to the

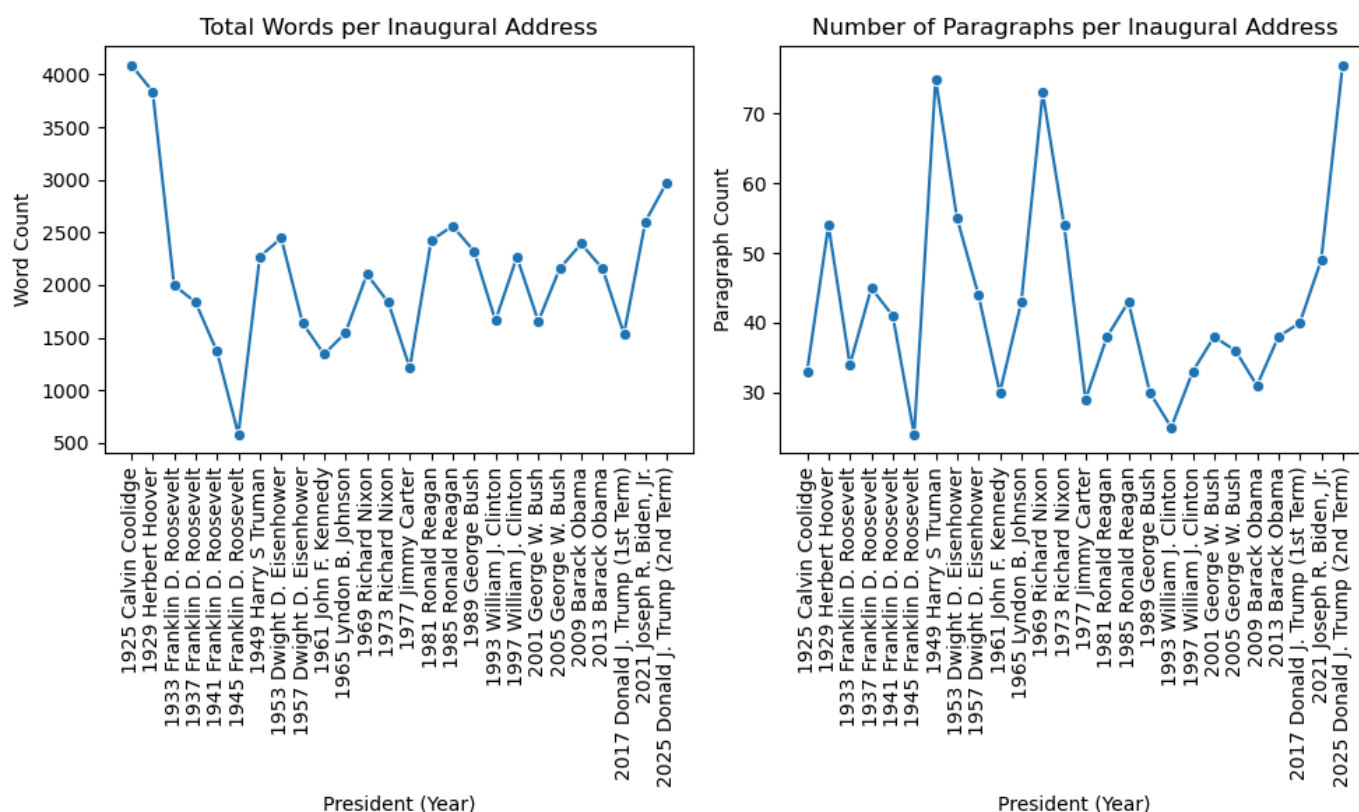
1. The American Presidency project <https://www.presidency.ucsb.edu/>

corresponding value from the Schwart original's 21-item questions used to measure these constructs. Details of the dictionary and the 21-item question can be found in the appendix. During the matching process, both the dictionary and the speech text were stemmed for higher chance of matching.

To measure an index for the values, relative word count percentage is applied to account for the variability of the number of words in each speech as shown in Figure 2. The formula for the dictionary approach value index is:

$$\text{Dictionary based value } X \text{ index} = \frac{\text{Number of matched words in dictionary value } X}{\text{Total number of word in speech}} \times 100\%$$

Figure 2: Number of words and number of paragraph of presidential inauguration speeches



To calculate the values index using the LLM approach, I used a locally hosted LLM model developed by Mistral. It is a 7B model that I have found to get a better nuance of the speech meaning than the popular Meta's llama 3.2(3B) or the similarly Alibaba's qwen(7B). The speech scraped in its raw format was already divided into short paragraphs. This is convenient as input for LLM. Using the whole speech as input would be too large of a context size and the model tends not to produce a good understanding of the speech. The prompt used to generate values for each paragraph is:

```
prompt = f"""You are an expert in political psychology. Classify the dominant human value expressed in this paragraph according to Schwartz's theory
```

\{"\{paragraph\}\{"\}

Respond with:

- 'openness' for Openness to Change,
- 'self-enhancement' for Self-Enhancement,
- 'conservation' for Conservation,
- 'self-transcendence' for Self-Transcendence,
- or 'neutral' if none apply.

Return one word only, paragraph's human value:

""

A key goal here is to make the LLM produce only the one value that was instructed.

Temperature of the model was set to 0.1 to reduce the chance the model gives unexpected results. Results outside of the expected values are set to 'Unknown'. Similarly to the dictionary approach, a relative proportion of values out of the total number of paragraphs was used to account for the variability of the number of paragraphs in each speech as shown in Figure 2. The formula for LLM values index of a speech is:

$$LLM \text{ based value } X \text{ index} = \frac{\text{Number of paragraphs labeled as } X}{\text{Total number of paragraphs in speech}} \times 100\%$$

The results for LLM-based and dictionary-based index will then be visualized and compared within and across parties with detailed case studies when mismatched are found.

Results

Figure 3 shows the word counts of the dictionary matched for each value over the last 100 years. The figure indicates that the dictionary does capture a good amount of word matching from the speeches. Figure 4 shows both the index of percentage word matched based on the dictionary approach and index of percentage of values using the LLM approach for Democrat's speeches. Figure 5 displays the same content but for the Republican.

For value openness, there was a rise in the mid 90s and then gradual decline in the Democrat presidential speech with the dictionary approach generally agreeing with LLM approach. This rise and fall pattern is also inline with my hypothesis. The explanation for this could be 1993 and 1997 under Clinton was a time of explosion in internet technology and embracement of diversity and changes. During Obama and Biden's time, there was a global financial crisis and large scale pandemic, which I think would make the speech focus less on openness value. With the Republican, there is no apparent trend throughout the 100 years period. There were some peaks in 1973 under Nixon and 2005 under Bush with the agreement of both LLM and dictionary methods. 2001 with Bush received a particularly low score from the LLM method relative to the dictionary method.

Figure 3: Number of words matched in the inauguration speech for each value

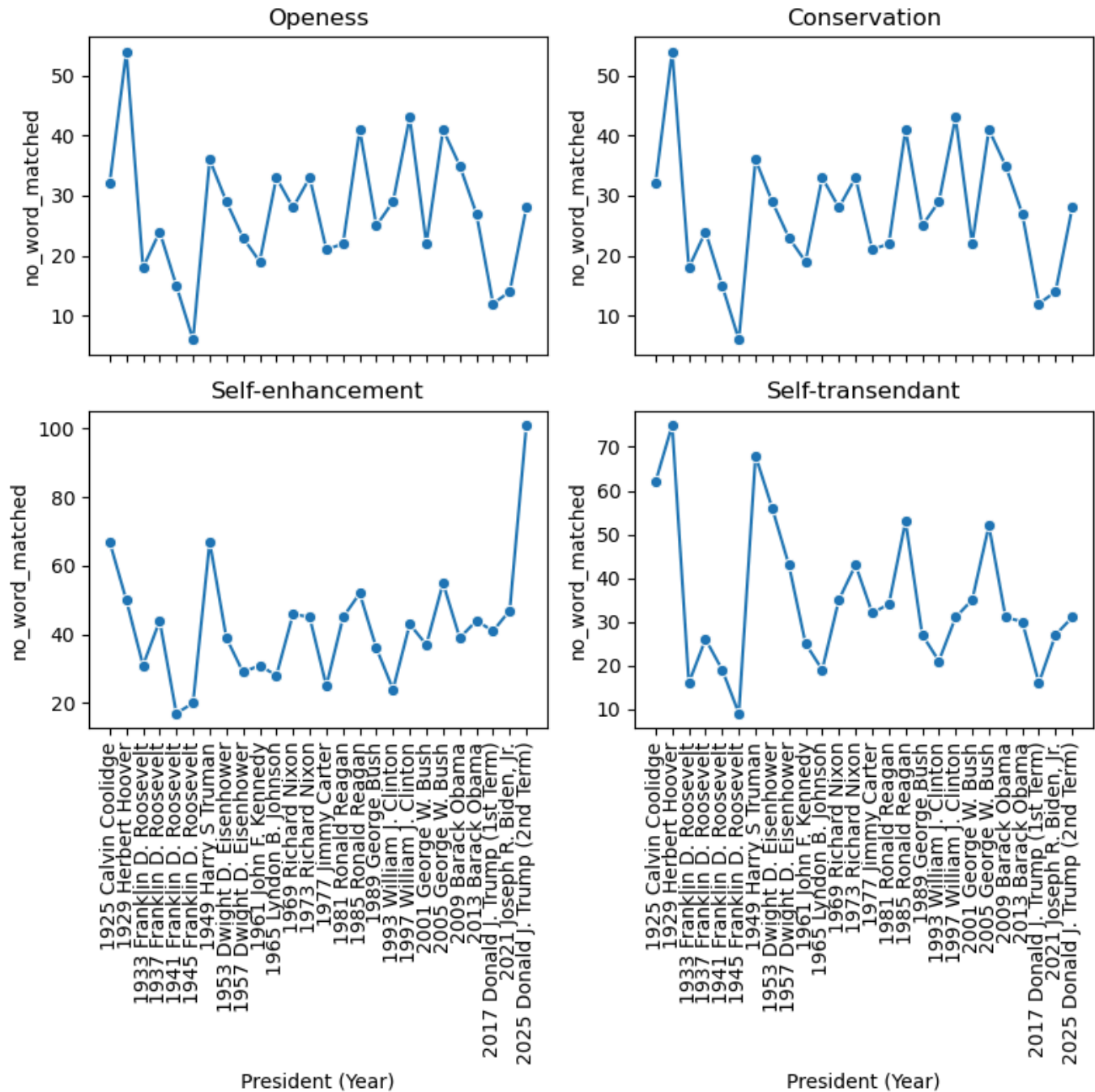


Figure 4: Democrats values based on dictionary and LLM method

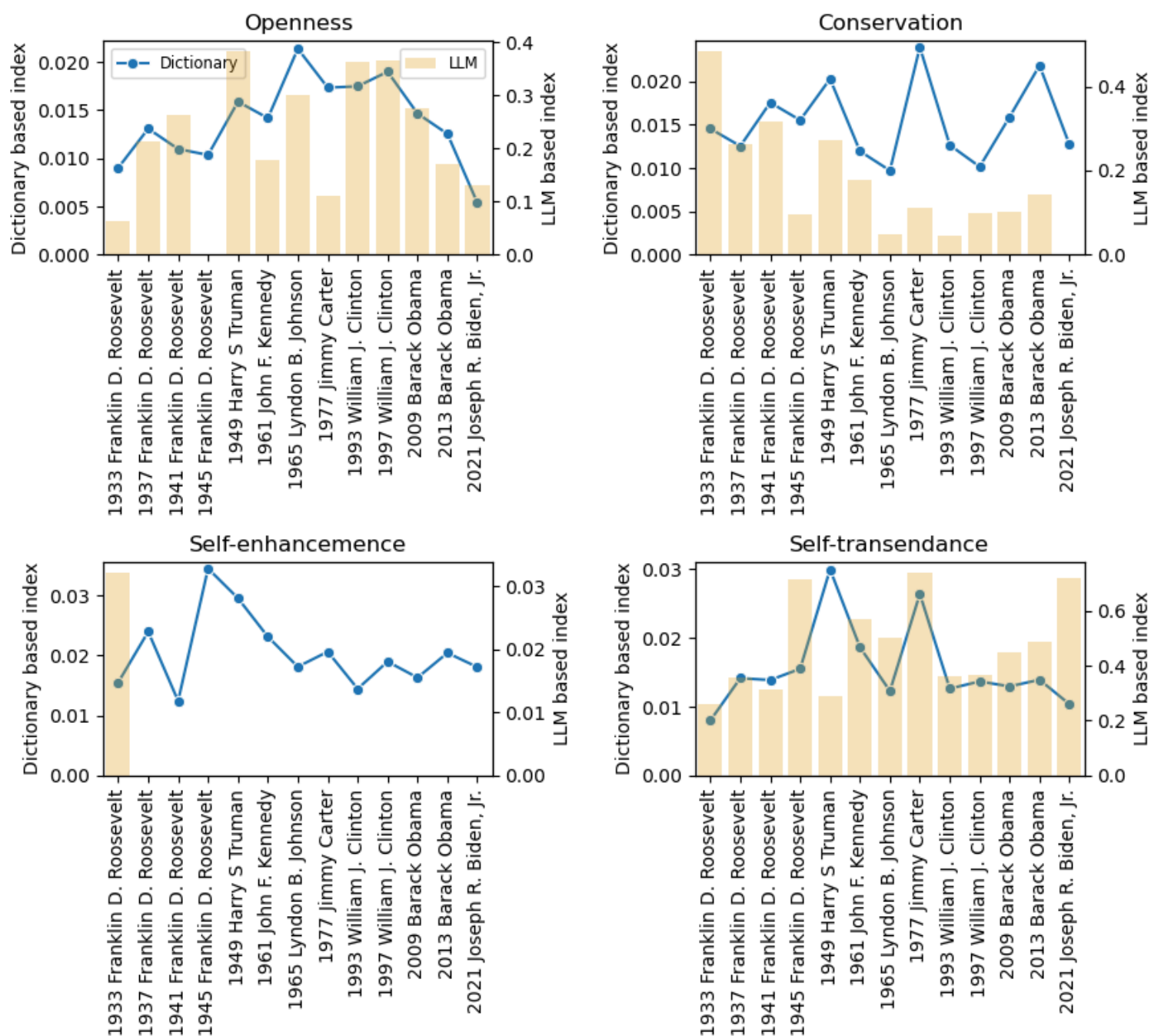
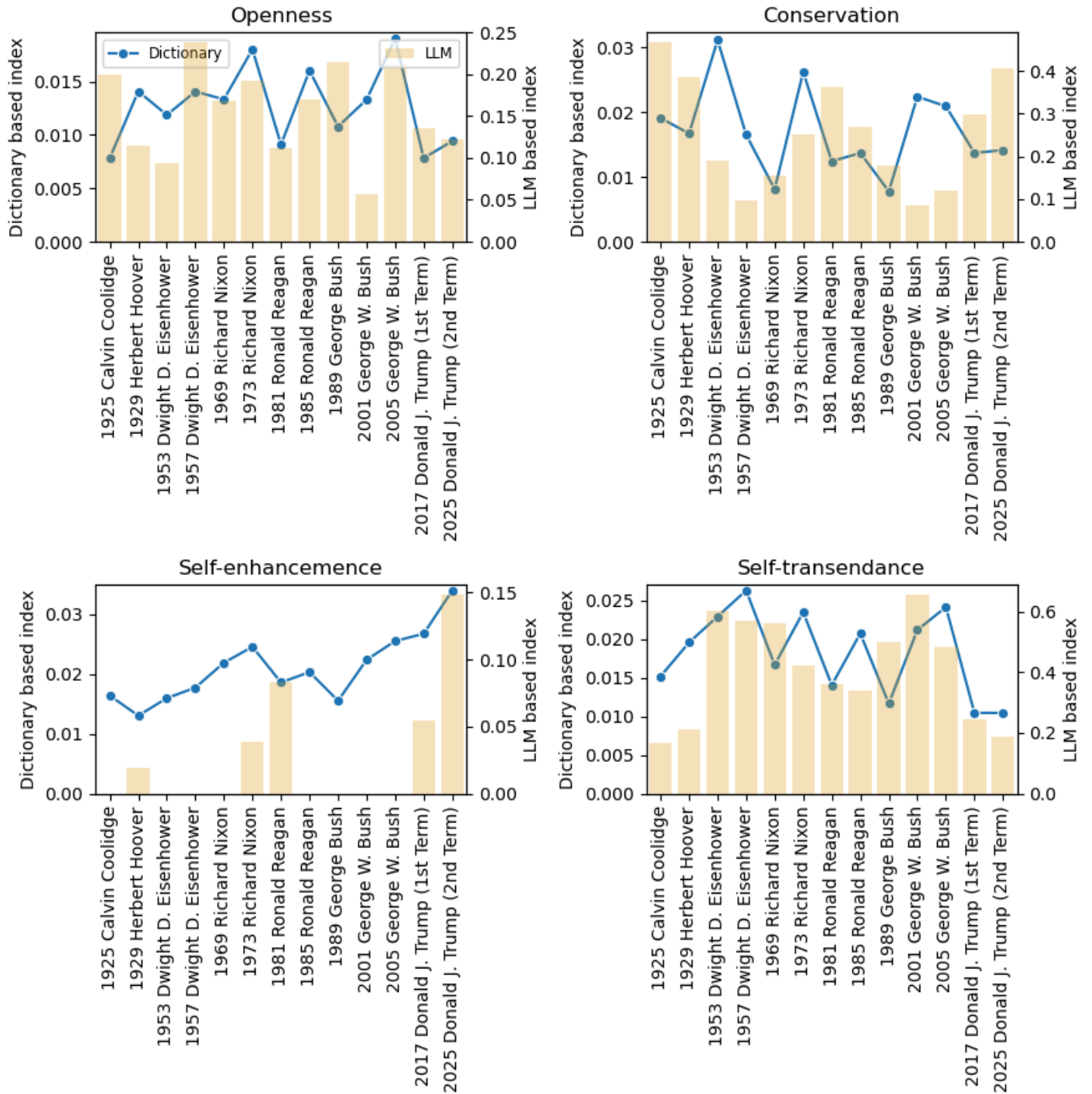


Figure 5: Republican values based on dictionary and LLM method



We should examine closely as a case study of the discrepancy between the two methods. There are only 2 out of 40 paragraphs that are classified as openness in Bush 2001 speech:

"Through much of the last century, America's faith in freedom and democracy was a rock in a raging sea. Now it is a seed upon the wind, taking root in many nations. Our democratic faith is more than the creed of our country. It is the inborn hope of our humanity, an ideal we carry but do not own, a trust we bear and pass along. Even after nearly 225 years, we have a long way yet to travel."

"While many of our citizens prosper, others doubt the promise, even the justice of our own country. The ambitions of some Americans are limited by failing schools and hidden prejudice and the circumstances of their birth. And sometimes our differences run so deep, it seems we share a continent but not a country. We do not accept this, and we will not allow it."

The words 'freedom' and 'hope' of the openness dictionary were identified in the first paragraph. I also agree with the general theme of openness classified by LLM Mistral. In the second paragraph, there are no matching words in the openness and some matching in the self-transcendence dictionary with the word 'justice' and 'share'. I disagree with Mistral and agree with the dictionary approach because I think the general theme is to disregard the lack of inclusion. In the reverse side, there are 2 paragraphs that contain the most matching with openness dictionary:

"Much time has passed since Jefferson arrived for his inauguration. The years and changes accumulate, but the themes of this day, he would know: our Nation's grand story of courage and its simple dream of dignity."

"America at its best is also courageous. Our national courage has been clear in times of depression and war, when defeating common dangers defined our common good. Now we must choose if the example of our fathers and mothers will inspire us or condemn us. We must show courage in a time of blessing by confronting problems instead of passing them on to future generations."

Both are classified as self-transcendence by LLM. I would classify the first one as openness and the second one as self-transcendence. This means the dictionary approach was more accurate in the first case and LLM was more accurate in the second case.

Till this point I think it is very difficult to know which approach is more accurate. Each has its own weakness and strength. The LLM seems to misclassify regularly but the dictionary approach would never be able to understand "confronting problems instead of passing them on to future generations" as self-transcendence like the LLM method does.

For the conservation values, we see a decline in these values in Democrats using LLM approach while dictionaries indicate a high variance straight line trend. The LLM approach here was more inline with my hypothesis. For the Republicans, the LLM approach suggests conservation was at its peak during 1925 Calvin Coolidge, 1981 Ronald Reagan and 2025 Donald J Trump. The dictionary also shows three peaks but at different time points: 1953 Eisenhower, 1973 Nixon and 2001 Bush.

Looking at the value of self-enhancement, there is a contrasting trend between the Democrats and Republicans after 1945 using a dictionary approach. Self-enhancement values in

Democrats experience a constant decline while Republican's is generally been increasing since 1925. Strangely, the LLM approach does not seem to be able to capture a lot of paragraphs with these values with most of the speech for both parties having no paragraph classified as self-enhancement. For Democrats, there is only one occasion where the value was found in 1933 Roosevelt.

"We do not distrust the future of essential democracy. The people of the United States have not failed. In their need they have registered a mandate that they want direct, vigorous action. They have asked for discipline and direction under leadership. They have made me the present instrument of their wishes. In the spirit of the gift I take it."

My judgement is that this paragraph contains both self-enhancement and conservation with its tone in power and discipline. The dictionary can identify conservation value by the word "discipline" but also found some openness matching by the word 'future' which is not very relevant here. Overall I think the LLM did a better job in this occasions. Due to my methodology design, the LLM can only give one value to a paragraph which limits its nuance when facing a multi-theme paragraph. This could be the reason why the LLM here seems not to be able to capture self-enhancement in most of the speeches because there are usually other themes mixed into the paragraph.

Self-transcendence is the value I expected to be increasing in both parties but does not seem to be true. According to the LLM approach the value has peaked three times for Democrats in 1945 Roosevelt, 1977 Carter and 2021 Biden. Republicans also showed two peak patterns at 1953 Eisenhower and 2001 Bush using LLM index. Dictionary approach on the other hand shows a decreasing trend in Republicans's speeches. Once again we see major mismatch between two approaches in 2021 Biden with LLM showing peak while dictionary indicates one of the lowest ever recorded. Some of the occasions where LLM classify paragraph as self-transcendence while dictionary detect no word matching for the same value is:

"In another January on New Year 's Day in 1863, Abraham Lincoln signed the Emancipation Proclamation. When he put pen to paper, the President said, and I quote, "If my name ever goes down into history it will be for this act, and my whole soul is in it." My whole soul is in it."

"Here we stand across the Potomac from Arlington Cemetery, where heroes who gave the last full measure of devotion rest in eternal peace."

I agree with both of the classification of the LLM. The first paragraph mentions the Emancipation Proclamation which is about ending the slavery in America while the second paragraph's main idea is about the sacrifice of the heroes. In this instance I have to give the LLM some edge as the dictionary approach would never be able to capture this due to the specificity of the word. My initial hypothesis about rising Self-transcendence can still be partly true if we only look at the LLM index of Democrats from 1993 on.

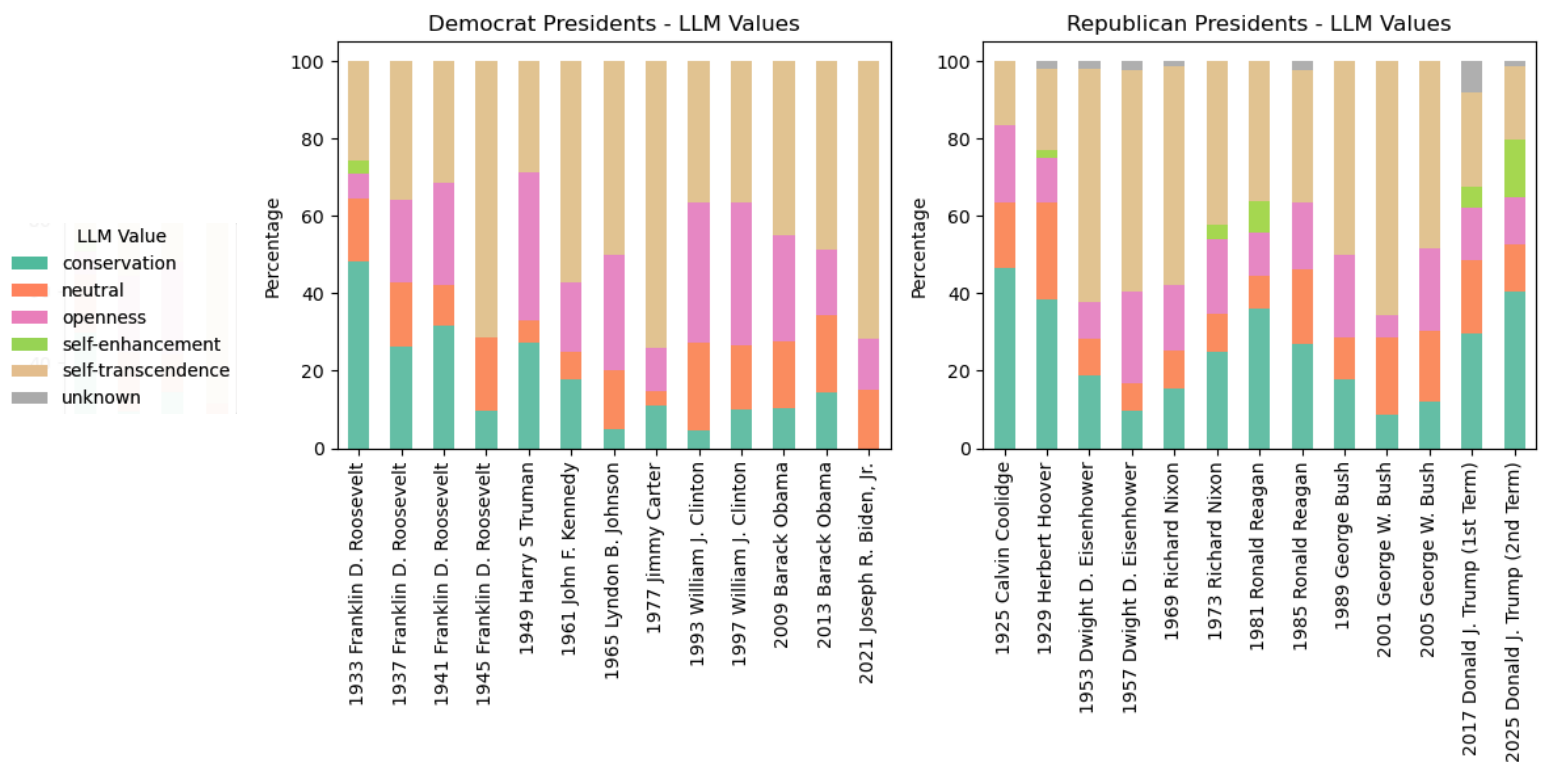
Discussion

Overall, based on different approaches we can see different patterns emerge from the evolution of Schwartz's values in the American presidential inauguration speeches. However, through a number of paragraph case studies, I would give the edge to the LLM approach. This is definitely not a rigorous final conclusion due to a small number of case studies and my personal

judgement of the theme of the paragraphs. It would be much better if we had some ground truth from a body of experts. The advantage of the dictionary approach is its agile and simple execution. While the LLM approach takes a lot more time to set up and execute. Due to this nature of the LLM approach, I have only required it to give only one value per paragraph. A better but much more time consuming approach would be to ask 4 questions for each value for each paragraph, requiring the LLM to rate how the paragraph contains the tone of each of the values in a score from 1 to 5.

Figure 4 and 5 focus much more on the comparison between 2 approaches and it is difficult to have an overview of the values evolution. To summarize clearly this transformation of American values, Figure 6 shows a stacked columns chart of the values for Republican and Democrats over the last 100 years using the LLM approach (since I have established that it is a more reliable approach). For the Democrats, self-transcendence has always been the dominant value, with a small mixing of openness. Conservation value used to be dominant in 1933 Roosevelt but quickly declined to the point of non-existence in 2021 Biden. The Republican values trend is more like a battleground of conservation and self-transcendence value. Openness is also mixed in but to a smaller extent than Democrats. Self-enhancement rarely emerges from any speech any time but there is a resurgence of this value in the last two Trump administration terms.

Figure 6: The evolution of the American's values by Democrats and Republicans



Appendix

Table of 21 questions measuring Schwartz's human values

Higher order value	Lower Order Human Value	Variable
Openness to change	Self-Direction	ipcrtiv: Important to think new ideas and be creative.
		impfree: Important to make own decisions and be free.
	Stimulation	impdiff: Important to try new and different things in life.
		ipadvnt: Important to seek adventures and have an exciting life.
	Hedonism (Partial)	ipgdtim: Important to have a good time.
		impfun: Important to seek fun and things that give pleasure.
Self enhancement	Hedonism (Partial)	ipgdtim: Important to have a good time.
		impfun: Important to seek fun and things that give pleasure.
	Achievement	ipshabt: Important to show abilities and be admired.
		ipsuces: Important to be successful and that people recognize achievements.
	Power	imprich: Important to be rich, have money, and expensive things.
		iprspot: Important to get respect from others.
Conservation	Security	impsafe: Important to live in secure and safe surroundings.
		ipstrgv: Important that government is strong and ensures safety.
	Conformity	ipfrule: Important to do what is told and follow rules.
		ipbhprp: Important to behave properly.
	Tradition	imptrad: Important to follow traditions and customs.
		ipmodst: Important to be humble and modest, not draw attention.
Self-transcendence	Benevolence	iphlppl: Important to help people and care for others' well-being.
		iplylfr: Important to be loyal to friends and devote to people close.
	Universalism	ipeqopt: Important that people are treated equally and have equal opportunities.
		ipudrst: Important to understand different people.
		impenv: Important to care for nature and environment.

Reference

1. Kubát, Miroslav & Čech, Radek. (2016). Quantitative Analysis of US Presidential Inaugural Addresses. *Glottometrics*. 15. 14-27.
2. Light, R. (2014). From Words to Networks and Back: Digital Text, Computational Social Science, and the Case of Presidential Inaugural Addresses. *Social Currents*, 1(2), 111-129. <https://doi.org/10.1177/2329496514524543> (Original work published 2014)
3. Shalom H. Schwartz, (1992) Universals in the Content and Structure of Values: Theoretica Advances and Empirical Tests in 20 Countries
4. Stuckey, M. (2010). Inaugural addresses. In *Encyclopedia of the American presidency* (Vol. 1, pp. 260-262). New York, NY: Infobase Publishing.
5. West, Victoria, "A Style of His Own: A Rhetorical Analysis of President Barack Obama's Inaugural Addresses" (2014). Undergraduate Research Awards. 21. <https://digitalcommons.hollins.edu/researchawards/21>
6. Zhou, Z. (2023). Analysis of Donald Trump's Inaugural Speech based on Attitude Within Appraisal Theory. *Journal of Education and Educational Research*, 4(1), 151-158. <https://doi.org/10.54097/jeer.v4i1.10275>

The sessions below are dedicated to the code used to produce the result of this assignment

QTA Code

April 23, 2025

```
[266]: from concurrent.futures import ThreadPoolExecutor
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time
import matplotlib.pyplot as plt
from datetime import datetime
from nltk.stem import SnowballStemmer
import nltk
nltk.download('punkt')

stemmer = SnowballStemmer("english")
#define dictionary
openness_to_change = [
    "freedom", "liberty", "innovation", "change", "progress",
    "reform", "future", "creativity", "dream", "aspiration",
    "hope", "new", "opportunity", "vision", "imagination",
    "bold", "renewal", "transformation", "explore", "possibility",
    "potential", "advancement", "modern", "shift", "trailblaze",
    "lead", "breakthrough", "expand", "dynamic", "revitalize",
    "fresh", "unleash", "rethink", "reimagine", "emerge",
    "reinvent", "courage", "pioneer", "initiative", "path",
    "discover", "awakening", "creative", "unfold", "novel",
    "inspire", "movement", "exploration", "adapt", "open",
    'adventure', 'exciting', 'fun', 'pleasure' #word from 21 item question
]

self_enhancement = [
    "power", "success", "strength", "leadership", "greatness",
    "achievement", "ambition", "victory", "win", "glory",
    "dominance", "control", "command", "triumph", "competitive",
    "drive", "excel", "pursuit", "goal", "influence",
    "mastery", "authority", "prestige", "excellence", "merit",
    "fame", "recognition", "courage", "boldness", "resolve",
    "fearless", "determination", "capacity", "build",
    "forge", "conquer", "expand", "secure", "rise",
```

```

    "strong", "heroic", "forward", "lead",
    "milestone", "achievement", "command", "focus", "target",
    "fun" , "pleasure", "admire", "ability", "rich", "money", "wealth" #word from 21 item question
]

conservation = [
    "tradition", "heritage", "faith", "family", "God",
    "order", "duty", "responsibility", "security", "protect",
    "defend", "preserve", "stability", "unity", "honor",
    "law", "custom", "discipline", "foundation", "obedience",
    "country", "homeland", "loyalty", "legacy", "sacrifice",
    "endure", "patriotism", "culture", "restore", "guard",
    "principle", "obedience", "respect", "safety",
    "defense", "maintain", "faithful", "uphold", "sacred",
    "integrity", "traditions", "belong", "structure", "moral",
    "stable", "preservation", "commitment", "commandment", "reverence",
    "safe", "ensure", "rule", "tradition", "humble", "modest" #word from 21 item question
]

self_transcendence = [
    "justice", "equality", "compassion", "peace", "kindness",
    "care", "dignity", "unity", "inclusion", "solidarity",
    "humanity", "service", "respect", "rights", "freedom",
    "tolerance", "empathy", "support", "cooperation", "altruism",
    "help", "uplift", "share", "heal", "welfare",
    "together", "bridge", "hope", "forgive", "aid",
    "nurture", "generosity", "civility", "brotherhood", "neighbor",
    "liberty", "reconciliation", "open", "grace", "community",
    "embrace", "benevolence", "equal", "mutual",
    "serve", "affirm", "humility", "moral", "voice",
    "wellbeing", "nature", "environment" #word from 21 item question
]

#stem dictionary
stemmed_openness = [stemmer.stem(word) for word in openness_to_change]
stemmed_self_enhancement = [stemmer.stem(word) for word in self_enhancement]
stemmed_conservation = [stemmer.stem(word) for word in conservation]
stemmed_self_transcendence = [stemmer.stem(word) for word in self_transcendence]

#function rate text using qwen
def rate_paragraph_ideology_qwen(paragraph, model="qwen2.5", base_url="http://localhost:11434"):
    prompt = f"""You are an expert in political psychology. Classify the
    dominant human value expressed in this paragraph according to Schwartz's
    theory:
    \"{paragraph}\""""

```


Respond with:

- 'openness' for Openness to Change,
- 'self-enhancement' for Self-Enhancement,
- 'conservation' for Conservation,
- 'self-transcendence' for Self-Transcendence,
- or 'neutral' if none apply.

Return one word only, paragraph's human value:

"""

```
api_generate = f"{base_url}/api/generate"
payload = {
    "model": model,
    "prompt": prompt,
    "stream": False,
    "options": {
        "temperature": 0.0 # Recommended for classification
    }
}

try:
    response = requests.post(api_generate, json=payload)
    response.raise_for_status()
    data = response.json()

    # Clean and parse the output
    output = data.get("response", "").strip()
    if output.lower() in ["openness", "self-enhancement", "conservation", "self-transcendence", "neutral"]:
        return output.lower()
    else:
        print(f"[WARN] Unexpected response content: {output}")
        return 'Unknown'
except Exception as e:
    print(f"[ERROR] LLM error for model {model}: {e}")
    return 0

#function rate text using other model
def rate_paragraph_ideology(paragraph,model):
    prompt = f"""You are an expert in political psychology. Classify the
    dominant human value expressed in this paragraph according to Schwartz's
    theory:
    \"{paragraph}\"
```

Respond with:

- 'openness' for Openness to Change,
- 'self-enhancement' for Self-Enhancement,
- 'conservation' for Conservation,
- 'self-transcendence' for Self-Transcendence,
- or 'neutral' if none apply.

Return one word only, paragraph's human value:

"""

```
try:
    res = requests.post(
        "http://localhost:11434/api/generate",
        json={
            "model": model,
            "prompt": prompt,
            "stream": False,
            "options": {
                "temperature": 0
            }
        }
    )
    score_str = res.json()["response"].strip()
    return score_str.lower() if score_str.lower() in ["openness",
↪ "self-enhancement", "conservation", "self-transcendence", "neutral"] else
↪ "unknown"
except Exception as e:
    print(f"LLM error: {e}")
    return 0

# Base URL pattern to scrap text
BASE_URL = 'https://www.presidency.ucsb.edu/documents/inaugural-address-{'
special_cases = {
    55: "https://www.presidency.ucsb.edu/documents/second-inaugural-address",
    56: "https://www.presidency.ucsb.edu/documents/
↪ the-presidents-inaugural-address",
    57: "https://www.presidency.ucsb.edu/documents/third-inaugural-address",
    58: "https://www.presidency.ucsb.edu/documents/
↪ oath-office-and-second-inaugural-address",
    59: 'https://www.presidency.ucsb.edu/documents/inaugural-address'
}

# Function to stem text
def stem_text(text):
    words = text.lower().split()
```

```

    return [stemmer.stem(word) for word in words]

# List to hold paragraph-level data
speech_data = []

# Loop through the document IDs and fetch their data
for doc_id in range(0, 60): # Extend range as needed
    url = special_cases.get(doc_id, BASE_URL.format(doc_id)) # fallback to
    ↪BASE_URL if not special
    try:
        print(f"Fetching {url}")
        res = requests.get(url)
        if res.status_code != 200:
            print(f"Skipped {doc_id} (HTTP {res.status_code})")
            continue

        soup = BeautifulSoup(res.text, 'html.parser')
        date_tag = soup.select_one('span.date-display-single')
        date_iso = date_tag.get("content") if date_tag and date_tag.
        ↪has_attr("content") else "Unknown"

        try:
            # Extract year from the date (only if valid date)
            year = datetime.fromisoformat(date_iso).year
            # Skip if the year is less than 1925
            if year < 1925:
                print(f"Skipping {doc_id} (Year {year} is before 1925)")
                continue
        except ValueError:
            print(f"Skipping {doc_id} (Invalid date format)")
            continue

        # Extract paragraphs and full text
        paragraphs = []
        for p in soup.find_all("p"):
            text = p.get_text(strip=True)
            if text.startswith("Note:"):
                break # Stop appending after "Note:"
            paragraphs.append(text)

        # Extract president information
        pres_tag = soup.select_one('a[href^="/people/president/"]')
        president = pres_tag.get_text(strip=True) if pres_tag else "Unknown"

        # Loop through paragraphs and store individual paragraph data
        for para_idx, paragraph in enumerate(paragraphs):

```

```

        with ThreadPoolExecutor() as executor:
            future_qwen = executor.submit(rate_paragraph_ideology_qwen,
↪paragraph)
            future_mistral = executor.submit(rate_paragraph_ideology,
↪paragraph, "mistral")
            llm_value_qwen = future_qwen.result()
#            llm_value_mistral = future_mistral.result()
            llm_value_mistral = rate_paragraph_ideology( paragraph, "mistral")
            stemmed_text = stem_text(paragraph)
            openness_count = sum(word in stemmed_text for word in
↪stemmed_openness)
            conservation_count = sum(word in stemmed_text for word in
↪stemmed_conservation)
            self_en_count = sum(word in stemmed_text for word in
↪stemmed_self_enhancement)
            self_trancd_count = sum(word in stemmed_text for word in
↪stemmed_self_transcendence)
            openness_dict = {word: stemmed_text.count(word) for word in
↪stemmed_openness if stemmed_text.count(word) > 0}
            conservation_dict = {word: stemmed_text.count(word) for word in
↪stemmed_conservation if stemmed_text.count(word) > 0}
            self_en_dict = {word: stemmed_text.count(word) for word in
↪stemmed_self_enhancement if stemmed_text.count(word) > 0}
            self_trancd_dict = {word: stemmed_text.count(word) for word in
↪stemmed_self_transcendence if stemmed_text.count(word) > 0}

        speech_data.append({
            "doc_id": doc_id,
            "president": president,
            "year": year,
            "paragraph_idx": para_idx + 1, # Index the paragraph
            "openness_count": openness_count,
            "conservation_count": conservation_count,
            "self_en_count": self_en_count,
            "self_trancd_count": self_trancd_count,
            "llm_value_mistral": llm_value_mistral,
#            "llm_value_qwen": llm_value_qwen,
            "openness_dict" : openness_dict,
            "conservation_dict" : conservation_dict,
            "self_en_dict" : self_en_dict,
            "self_trancd_dict" : self_trancd_dict,
            "text": paragraph
        })

```

```

time.sleep(1)  # Be polite to the server

except Exception as e:
    print(f"Error processing doc {doc_id}: {e}")
    continue

# Save the data to a CSV
df_dict_only = pd.DataFrame(speech_data)
df.to_csv("inaugural_addresses_paragraph_level_fin_4.csv", index=False)
print(" Done! Saved to inaugural_addresses_paragraph_level.csv")

```

[nltk_data] Downloading package punkt to /Users/tpminh/nltk_data...

[nltk_data] Package punkt is already up-to-date!

```

Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-0
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-1
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-2
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-3
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-4
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-5
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-6
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-7
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-8
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-9
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-10
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-11
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-12
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-13
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-14
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-15
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-16
Skipping 16 (Year 1789 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-17
Skipping 17 (Year 1793 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-18
Skipping 18 (Year 1797 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-19
Skipping 19 (Year 1801 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-20
Skipping 20 (Year 1805 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-21
Skipping 21 (Year 1809 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-22
Skipping 22 (Year 1813 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-23
Skipping 23 (Year 1817 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-24
Skipping 24 (Year 1821 is before 1925)

```

Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-25>
Skipping 25 (Year 1825 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-26>
Skipping 26 (Year 1829 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-27>
Skipping 27 (Year 1833 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-28>
Skipping 28 (Year 1837 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-29>
Skipping 29 (Year 1841 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-30>
Skipping 30 (Year 1845 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-31>
Skipping 31 (Year 1849 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-32>
Skipping 32 (Year 1853 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-33>
Skipping 33 (Year 1857 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-34>
Skipping 34 (Year 1861 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-35>
Skipping 35 (Year 1865 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-36>
Skipping 36 (Year 1869 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-37>
Skipping 37 (Year 1873 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-38>
Skipping 38 (Year 1877 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-39>
Skipping 39 (Year 1881 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-40>
Skipping 40 (Year 1885 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-41>
Skipping 41 (Year 1889 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-42>
Skipping 42 (Year 1893 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-43>
Skipping 43 (Year 1897 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-44>
Skipping 44 (Year 1901 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-45>
Skipping 45 (Year 1905 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-46>
Skipping 46 (Year 1909 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-47>
Skipping 47 (Year 1913 is before 1925)
Fetching <https://www.presidency.ucsb.edu/documents/inaugural-address-48>
Skipping 48 (Year 1917 is before 1925)


```

Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-49
Skipping 49 (Year 1921 is before 1925)
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-50
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-51
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-52
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-53
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address-54
Fetching https://www.presidency.ucsb.edu/documents/second-inaugural-address
Fetching https://www.presidency.ucsb.edu/documents/the-presidents-inaugural-address
Fetching https://www.presidency.ucsb.edu/documents/third-inaugural-address
Fetching https://www.presidency.ucsb.edu/documents/oath-office-and-second-inaugural-address
Fetching https://www.presidency.ucsb.edu/documents/inaugural-address
Done! Saved to inaugural_addresses_paragraph_level.csv

```

```

[218]: # President-party dictionary
president_party = {
    "George Washington": "Independent",
    "John Adams": "Federalist",
    "Thomas Jefferson": "Democratic-Republican",
    "James Madison": "Democratic-Republican",
    "James Monroe": "Democratic-Republican",
    "John Quincy Adams": "Democratic-Republican",
    "Andrew Jackson": "Democrat",
    "Martin Van Buren": "Democrat",
    "William Henry Harrison": "Whig",
    "John Tyler": "Whig",
    "James Knox Polk": "Democrat",
    "Zachary Taylor": "Whig",
    "Millard Fillmore": "Whig",
    "Franklin Pierce": "Democrat",
    "James Buchanan": "Democrat",
    "Abraham Lincoln": "Republican",
    "Andrew Johnson": "National Union",
    "Ulysses S. Grant": "Republican",
    "Rutherford B. Hayes": "Republican",
    "James A. Garfield": "Republican",
    "Chester A. Arthur": "Republican",
    "Grover Cleveland": "Democrat",
    "Benjamin Harrison": "Republican",
    "William McKinley": "Republican",
    "Theodore Roosevelt": "Republican",
    "William Howard Taft": "Republican",
    "Woodrow Wilson": "Democrat",
    "Warren G. Harding": "Republican",
    "Calvin Coolidge": "Republican",
}

```

```

    "Herbert Hoover": "Republican",
    "Franklin D. Roosevelt": "Democrat",
    "Harry S Truman": "Democrat",
    "Dwight D. Eisenhower": "Republican",
    "John F. Kennedy": "Democrat",
    "Lyndon B. Johnson": "Democrat",
    "Richard Nixon": "Republican",
    "Gerald Ford": "Republican",
    "Jimmy Carter": "Democrat",
    "Ronald Reagan": "Republican",
    "George Bush": "Republican",
    "William J. Clinton": "Democrat",
    "George W. Bush": "Republican",
    "Barack Obama": "Democrat",
    "Donald J. Trump (1st Term)": "Republican",
    "Donald J. Trump (2nd Term)": "Republican",
    "Joseph R. Biden, Jr.": "Democrat"
}

```

```
[219]: df['party'] = df['president'].map(president_party)
```

```

/var/folders/np/yjbf57612_30_mjbr_d794yh0000gn/T/ipykernel_68929/1228184540.py:1
: SettingWithCopyWarning:

```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
`df['party'] = df['president'].map(president_party)`

```

[221]: #remove irrelevant data
df = df[(df['text'] != 'AboutSearch') & (df['text'] != '')]

```

```
[220]: df[df['party'].isna()]
```

```

[220]: Empty DataFrame
Columns: [doc_id, president, year, paragraph_idx, openness_count,
conservation_count, self_en_count, self_trancd_count, llm_value_mistral, text,
word_count, president_year, party]
Index: []

```

```
[303]: df
```

```

[303]:   doc_id  president  year  paragraph_idx  openness_count  \
2        0  Jimmy Carter  1977             3              0
3        0  Jimmy Carter  1977             4              1
4        0  Jimmy Carter  1977             5              1
5        0  Jimmy Carter  1977             6              1

```

6	0	Jimmy Carter	1977	7	5
...
1107	59	George Bush	1989	26	0
1108	59	George Bush	1989	27	0
1109	59	George Bush	1989	28	0
1110	59	George Bush	1989	29	2
1111	59	George Bush	1989	30	0

	conservation_count	self_en_count	self_transcd_count	\
2	0	0	1	
3	0	1	0	
4	1	0	1	
5	0	0	0	
6	2	2	0	
...	
1107	0	0	0	
1108	0	1	0	
1109	1	0	0	
1110	0	1	2	
1111	1	0	0	

	llm_value_mistral	text	\
2	self-transcendence	For myself and for our Nation, I want to thank...	
3	conservation	In this outward and physical ceremony, we atte...	
4	self-transcendence	Here before me is the Bible used in the inaugu...	
5	openness	This inauguration ceremony marks a new beginni...	
6	self-transcendence	Two centuries ago, our Nation's birth was a mi...	
...	
1107	openness	A President is neither prince nor pope, and I ...	
1108	conservation	There are few clear areas in which we as a soc...	
1109	self-transcendence	And so, there is much to do. And tomorrow the ...	
1110	self-transcendence	Some see leadership as high drama and the soun...	
1111	self-transcendence	Thank you. God bless you. And God bless the Un...	

	word_count	president_year	party	\
2	21	1977 Jimmy Carter	Democrat	
3	42	1977 Jimmy Carter	Democrat	
4	79	1977 Jimmy Carter	Democrat	
5	36	1977 Jimmy Carter	Democrat	
6	51	1977 Jimmy Carter	Democrat	
...	
1107	34	1989 George Bush	Republican	
1108	78	1989 George Bush	Republican	
1109	56	1989 George Bush	Republican	
1110	71	1989 George Bush	Republican	
1111	13	1989 George Bush	Republican	

```

openess_dict conservation_dict \
2          {}          {}
3      {'chang': 1}      {}
4      {'open': 1}      {'humbl': 1}
5      {'new': 4}      {}
6      {'dream': 2, 'new': 1, 'bold': 1, 'fresh': 1, ...      {'faith': 1}
...
1107          {}          {}
1108          {}          {}
1109          {}          {'god': 1}
1110      {'hope': 1, 'new': 1}      {}
1111          {}          {'god': 2}

self_en_dict      self_trancd_dict
2          {}      {'heal': 1}
3      {'strength': 1}      {}
4          {}      {'open': 1}
5          {}      {}
6      {'bold': 1, 'mileston': 1}      {}
...
1107          {}      {}
1108      {'rise': 1}      {}
1109          {}      {}
1110      {'leadership': 1}      {'hope': 1, 'generos': 1}
1111          {}      {}

[1046 rows x 17 columns]

```

```
[222]: # Word count per paragraph
df["word_count"] = df["text"].apply(lambda x: len(x.split()))
```

```
[223]: df['president_year'] = df['year'].astype(str) + " " + df['president']
```

```
[269]: df_groupby = df.groupby(['president_year', 'party']).agg(no_word =
    ↪ ('word_count', 'sum'), no_paragraph = ('paragraph_idx', 'count') ).
    ↪ reset_index().sort_values('president_year')

manual = df.groupby("president_year")[["openess_count", "conservation_count",
    ↪ "self_en_count", "self_trancd_count"]].sum()

# Total words per year
manual["total_words"] = df.groupby("president_year")["word_count"].sum()

manual_dict = manual[["openess_count", "conservation_count", "self_en_count",
    ↪ "self_trancd_count"]].div(manual["total_words"], axis=0).reset_index().
    ↪ sort_values('president_year')
```

```

llm_counts = df.groupby(['president_year', 'llm_value_mistral']).size().
↳unstack(fill_value=0)
llm_pct = llm_counts.div(llm_counts.sum(axis=1), axis=0).reset_index().
↳sort_values('president_year')

# Percentage of each value category per total words in speech
df_groupby[["openess_perc", "conservation_perc", "self_en_perc",
↳"self_tranecd_perc"]] = manual_dict[["openess_count", "conservation_count",
↳"self_en_count", "self_tranecd_count"]]

df_groupby[["openess_count", "conservation_count", "self_en_count",
↳"self_tranecd_count"]] = manual.reset_index()[["openess_count",
↳"conservation_count", "self_en_count", "self_tranecd_count"]]

df_groupby[['llm_conservation', 'llm_openess',
↳'llm_selfenhancement', 'llm_selftranscendence' ]] =
↳llm_pct[['conservation', 'openness', 'self-enhancement', 'self-transcendence']]

```

[228]: df_groupby

[228]:

	president_year	party	no_word	no_paragraph	\
0	1925 Calvin Coolidge	Republican	4091	30	
1	1929 Herbert Hoover	Republican	3838	52	
2	1933 Franklin D. Roosevelt	Democrat	1996	31	
3	1937 Franklin D. Roosevelt	Democrat	1836	42	
4	1941 Franklin D. Roosevelt	Democrat	1368	38	
5	1945 Franklin D. Roosevelt	Democrat	580	21	
6	1949 Harry S Truman	Democrat	2269	73	
7	1953 Dwight D. Eisenhower	Republican	2445	53	
8	1957 Dwight D. Eisenhower	Republican	1637	42	
9	1961 John F. Kennedy	Democrat	1341	28	
10	1965 Lyndon B. Johnson	Democrat	1545	40	
11	1969 Richard Nixon	Republican	2102	71	
12	1973 Richard Nixon	Republican	1833	52	
13	1977 Jimmy Carter	Democrat	1212	27	
14	1981 Ronald Reagan	Republican	2422	36	
15	1985 Ronald Reagan	Republican	2560	41	
16	1989 George Bush	Republican	2320	28	
17	1993 William J. Clinton	Democrat	1661	22	
18	1997 William J. Clinton	Democrat	2266	30	
19	2001 George W. Bush	Republican	1654	35	
20	2005 George W. Bush	Republican	2156	33	
21	2009 Barack Obama	Democrat	2391	29	
22	2013 Barack Obama	Democrat	2154	35	
23	2017 Donald J. Trump (1st Term)	Republican	1534	37	
24	2021 Joseph R. Biden, Jr.	Democrat	2602	46	
25	2025 Donald J. Trump (2nd Term)	Republican	2975	74	

	openess_perc	conservation_perc	self_en_perc	self_trancd_perc	\
0	0.007822	0.019066	0.016377	0.015155	
1	0.014070	0.016675	0.013028	0.019541	
2	0.009018	0.014529	0.015531	0.008016	
3	0.013072	0.012527	0.023965	0.014161	
4	0.010965	0.017544	0.012427	0.013889	
5	0.010345	0.015517	0.034483	0.015517	
6	0.015866	0.020273	0.029528	0.029969	
7	0.011861	0.031084	0.015951	0.022904	
8	0.014050	0.016494	0.017715	0.026268	
9	0.014169	0.011931	0.023117	0.018643	
10	0.021359	0.009709	0.018123	0.012298	
11	0.013321	0.008088	0.021884	0.016651	
12	0.018003	0.026187	0.024550	0.023459	
13	0.017327	0.023927	0.020627	0.026403	
14	0.009083	0.012386	0.018580	0.014038	
15	0.016016	0.013672	0.020313	0.020703	
16	0.010776	0.007759	0.015517	0.011638	
17	0.017459	0.012643	0.014449	0.012643	
18	0.018976	0.010150	0.018976	0.013680	
19	0.013301	0.022370	0.022370	0.021161	
20	0.019017	0.020872	0.025510	0.024119	
21	0.014638	0.015893	0.016311	0.012965	
22	0.012535	0.021820	0.020427	0.013928	
23	0.007823	0.013690	0.026728	0.010430	
24	0.005380	0.012683	0.018063	0.010377	
25	0.009412	0.014118	0.033950	0.010420	

	openess_count	conservation_count	self_en_count	self_trancd_count	\
0	32	78	67	62	
1	54	64	50	75	
2	18	29	31	16	
3	24	23	44	26	
4	15	24	17	19	
5	6	9	20	9	
6	36	46	67	68	
7	29	76	39	56	
8	23	27	29	43	
9	19	16	31	25	
10	33	15	28	19	
11	28	17	46	35	
12	33	48	45	43	
13	21	29	25	32	
14	22	30	45	34	
15	41	35	52	53	
16	25	18	36	27	

17	29	21	24	21
18	43	23	43	31
19	22	37	37	35
20	41	45	55	52
21	35	38	39	31
22	27	47	44	30
23	12	21	41	16
24	14	33	47	27
25	28	42	101	31

	llm_conservation	llm_openess	llm_selfenhancement	llm_selftranscendence
0	0.466667	0.200000	0.000000	0.166667
1	0.384615	0.115385	0.019231	0.211538
2	0.483871	0.064516	0.032258	0.258065
3	0.261905	0.214286	0.000000	0.357143
4	0.315789	0.263158	0.000000	0.315789
5	0.095238	0.000000	0.000000	0.714286
6	0.273973	0.383562	0.000000	0.287671
7	0.188679	0.094340	0.000000	0.603774
8	0.095238	0.238095	0.000000	0.571429
9	0.178571	0.178571	0.000000	0.571429
10	0.050000	0.300000	0.000000	0.500000
11	0.154930	0.169014	0.000000	0.563380
12	0.250000	0.192308	0.038462	0.423077
13	0.111111	0.111111	0.000000	0.740741
14	0.361111	0.111111	0.083333	0.361111
15	0.268293	0.170732	0.000000	0.341463
16	0.178571	0.214286	0.000000	0.500000
17	0.045455	0.363636	0.000000	0.363636
18	0.100000	0.366667	0.000000	0.366667
19	0.085714	0.057143	0.000000	0.657143
20	0.121212	0.212121	0.000000	0.484848
21	0.103448	0.275862	0.000000	0.448276
22	0.142857	0.171429	0.000000	0.485714
23	0.297297	0.135135	0.054054	0.243243
24	0.000000	0.130435	0.000000	0.717391
25	0.405405	0.121622	0.148649	0.189189

```
[270]: # Set figure and axis layout
fig, axes = plt.subplots(1, 2, figsize=(10, 6), sharex=True)

# Plot total words per speech
sns.lineplot(ax=axes[0], data=df_groupby, x='president_year', y='no_word',
             marker='o')
axes[0].set_title('Total Words per Inaugural Address')
axes[0].set_xlabel('President (Year)')
axes[0].set_ylabel('Word Count')
```

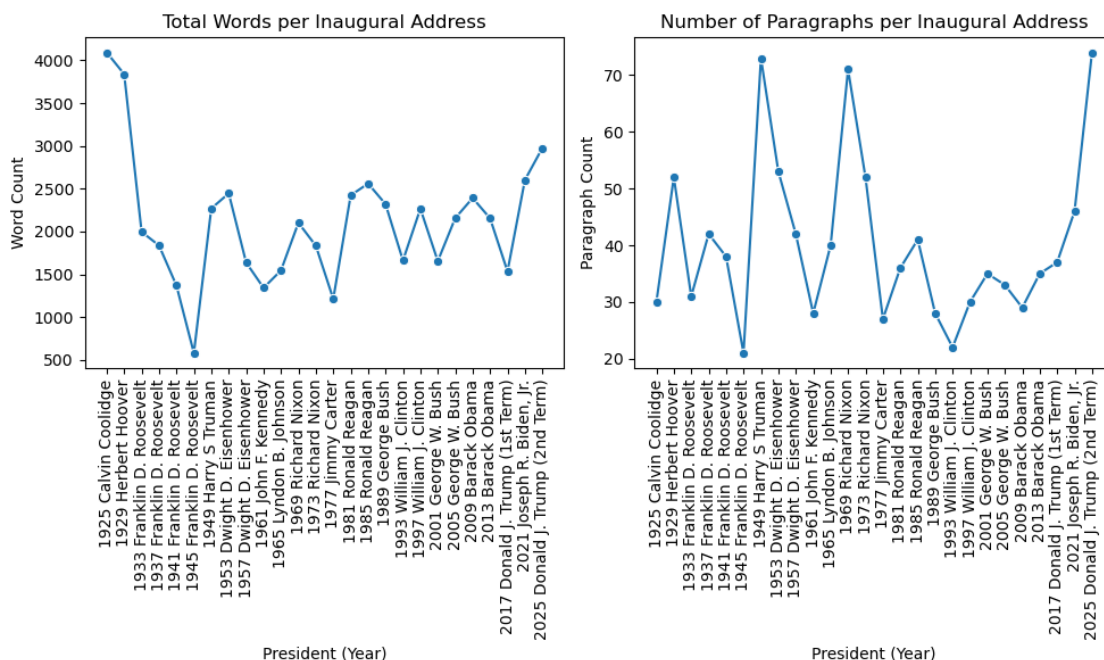
```

axes[0].tick_params(axis='x', rotation=90)

# Plot number of paragraphs per speech
sns.lineplot(ax=axes[1], data=df_groupby, x='president_year', y='no_paragraph',
             marker='o')
axes[1].set_title('Number of Paragraphs per Inaugural Address')
axes[1].set_xlabel('President (Year)')
axes[1].set_ylabel('Paragraph Count')
axes[1].tick_params(axis='x', rotation=90)

# Adjust layout
plt.tight_layout()
plt.show()

```



```

[271]: #plot values by all parties
# Set figure and axis layout
fig, axes = plt.subplots(2, 2, figsize=(12, 10), sharex=False)
axes = axes.flatten()
# Plot total words per speech

sns.lineplot(ax=axes[0], data=df_groupby, x='president_year', y='openness_perc',
             marker='o', label='Dictionary')
axes[0].set_title('Openness')
axes[0].set_xlabel('')
axes[0].set_ylabel('Dictionary based index')

```

```

axes[0].tick_params(axis='x', rotation=90)
ax02 = axes[0].twinx()
sns.barplot(ax=ax02, data=df_groupby, x='president_year', y='llm_openess',
    ↪alpha=0.3, color='orange', label='LLM')
axes[0].legend(loc='upper left', fontsize='small')
ax02.set_ylabel('LLM based index')
ax02.legend(loc='upper right', fontsize='small')

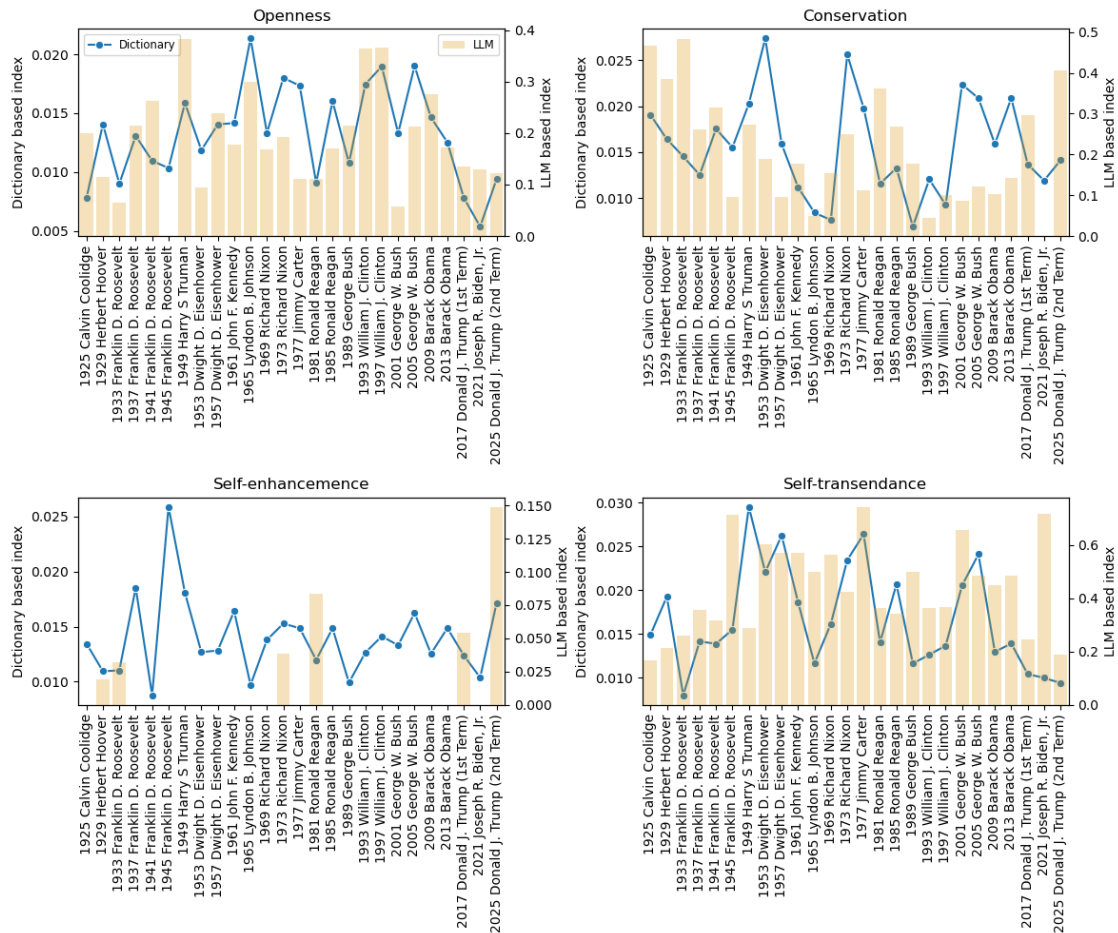
sns.lineplot(ax=axes[1], data=df_groupby, x='president_year',
    ↪y='conservation_perc', marker='o')
axes[1].set_title('Conservation')
axes[1].set_xlabel('')
axes[1].set_ylabel('Dictionary based index')
axes[1].tick_params(axis='x', rotation=90)
ax12 = axes[1].twinx()
sns.barplot(ax=ax12, data=df_groupby, x='president_year', y='llm_conservation',
    ↪alpha=0.3, color='orange')
ax12.set_ylabel('LLM based index')

# Plot number of paragraphs per speech
sns.lineplot(ax=axes[2], data=df_groupby, x='president_year', y='self_en_perc',
    ↪marker='o')
axes[2].set_title('Self-enhancement')
axes[2].set_xlabel('')
axes[2].set_ylabel('Dictionary based index')
axes[2].tick_params(axis='x', rotation=90)
ax22 = axes[2].twinx()
sns.barplot(ax=ax22, data=df_groupby, x='president_year',
    ↪y='llm_selfenhancement', alpha=0.3, color='orange')
ax22.set_ylabel('LLM based index')

sns.lineplot(ax=axes[3], data=df_groupby, x='president_year',
    ↪y='self_tranecd_perc', marker='o')
axes[3].set_title('Self-transendance')
axes[3].set_xlabel('')
axes[3].set_ylabel('Dictionary based index')
axes[3].tick_params(axis='x', rotation=90)
ax32 = axes[3].twinx()
sns.barplot(ax=ax32, data=df_groupby, x='president_year',
    ↪y='llm_selftranscendence', alpha=0.3, color='orange')
ax32.set_ylabel('LLM based index')

```

```
# Adjust layout
plt.tight_layout()
plt.show()
```



```
[272]: #plot values for democrats
df_party = df_groupby[ df_groupby['party'] == 'Democrat' ]

# Set figure and axis layout
fig, axes = plt.subplots(2, 2, figsize=(9, 8), sharex=False)
axes = axes.flatten()
# Plot total words per speech

sns.lineplot(ax=axes[0], data=df_party, x='president_year', y='openness_perc',
             marker='o', label='Dictionary')
axes[0].set_title('Openness')
axes[0].set_xlabel('')
axes[0].set_ylabel('Dictionary based index')
```

```

axes[0].tick_params(axis='x', rotation=90)
ax02 = axes[0].twinx()
sns.barplot(ax=ax02, data=df_party, x='president_year', y='llm_openess',
            alpha=0.3, color='orange', label='LLM')
axes[0].legend(loc='upper left', fontsize='small')
ax02.set_ylabel('LLM based index')
ax02.legend(loc='upper right', fontsize='small')
axes[0].set_ylim(bottom=0)

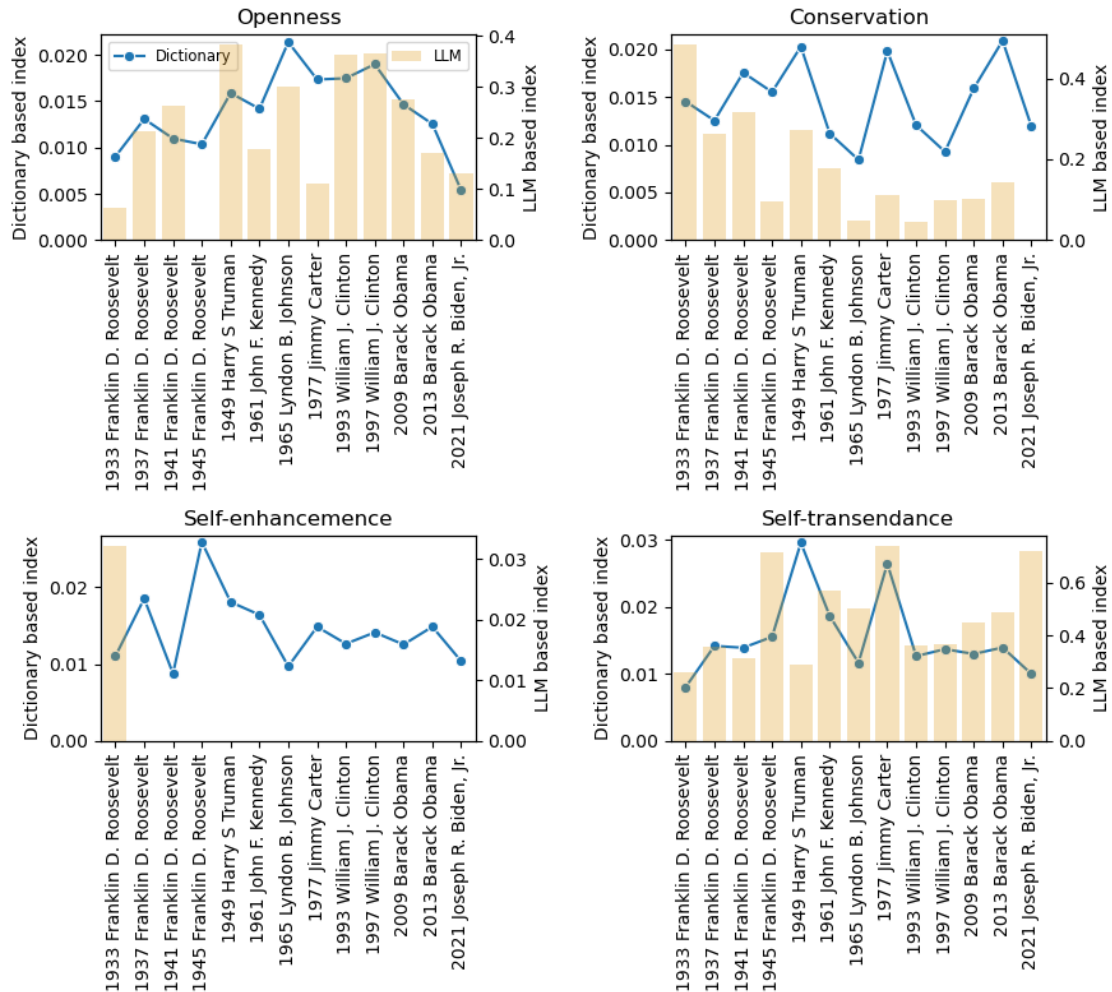
sns.lineplot(ax=axes[1], data=df_party, x='president_year',
            y='conservation_perc', marker='o')
axes[1].set_title('Conservation')
axes[1].set_xlabel('')
axes[1].set_ylabel('Dictionary based index')
axes[1].tick_params(axis='x', rotation=90)
ax12 = axes[1].twinx()
sns.barplot(ax=ax12, data=df_party, x='president_year', y='llm_conservation',
            alpha=0.3, color='orange')
ax12.set_ylabel('LLM based index')
axes[1].set_ylim(bottom=0)

# Plot number of paragraphs per speech
sns.lineplot(ax=axes[2], data=df_party, x='president_year', y='self_en_perc',
            marker='o')
axes[2].set_title('Self-enhancement')
axes[2].set_xlabel('')
axes[2].set_ylabel('Dictionary based index')
axes[2].tick_params(axis='x', rotation=90)
ax22 = axes[2].twinx()
sns.barplot(ax=ax22, data=df_party, x='president_year',
            y='llm_selfenhancement', alpha=0.3, color='orange')
ax22.set_ylabel('LLM based index')
axes[2].set_ylim(bottom=0)

sns.lineplot(ax=axes[3], data=df_party, x='president_year',
            y='self_tranecd_perc', marker='o')
axes[3].set_title('Self-transendance')
axes[3].set_xlabel('')
axes[3].set_ylabel('Dictionary based index')
axes[3].tick_params(axis='x', rotation=90)
ax32 = axes[3].twinx()
sns.barplot(ax=ax32, data=df_party, x='president_year',
            y='llm_selftranscendence', alpha=0.3, color='orange')
ax32.set_ylabel('LLM based index')
axes[3].set_ylim(bottom=0)

```

```
# Adjust layout
plt.tight_layout()
plt.show()
```



```
[273]: #plot values for republicans
```

```
df_party = df_groupby[ df_groupby['party'] == 'Republican' ]

# Set figure and axis layout
fig, axes = plt.subplots(2, 2, figsize=(9, 9), sharex=False)
axes = axes.flatten()
# Plot total words per speech

sns.lineplot(ax=axes[0], data=df_party, x='president_year', y='openness_perc',
             marker='o', label='Dictionary')
axes[0].set_title('Openness')
```



```

axes[0].set_xlabel('')
axes[0].set_ylabel('Dictionary based index')
axes[0].tick_params(axis='x', rotation=90)
ax02 = axes[0].twinx()
sns.barplot(ax=ax02, data=df_party, x='president_year', y='llm_openess',
    ↪alpha=0.3, color='orange', label='LLM')
axes[0].legend(loc='upper left', fontsize='small')
ax02.set_ylabel('LLM based index')
ax02.legend(loc='upper right', fontsize='small')
axes[0].set_ylim(bottom=0)

sns.lineplot(ax=axes[1], data=df_party, x='president_year',
    ↪y='conservation_perc', marker='o')
axes[1].set_title('Conservation')
axes[1].set_xlabel('')
axes[1].set_ylabel('Dictionary based index')
axes[1].tick_params(axis='x', rotation=90)
ax12 = axes[1].twinx()
sns.barplot(ax=ax12, data=df_party, x='president_year', y='llm_conservation',
    ↪alpha=0.3, color='orange')
ax12.set_ylabel('LLM based index')
axes[1].set_ylim(bottom=0)

# Plot number of paragraphs per speech
sns.lineplot(ax=axes[2], data=df_party, x='president_year', y='self_en_perc',
    ↪marker='o')
axes[2].set_title('Self-enhancement')
axes[2].set_xlabel('')
axes[2].set_ylabel('Dictionary based index')
axes[2].tick_params(axis='x', rotation=90)
ax22 = axes[2].twinx()
sns.barplot(ax=ax22, data=df_party, x='president_year',
    ↪y='llm_selfenhancement', alpha=0.3, color='orange')
ax22.set_ylabel('LLM based index')
axes[2].set_ylim(bottom=0)

sns.lineplot(ax=axes[3], data=df_party, x='president_year',
    ↪y='self_tranecd_perc', marker='o')
axes[3].set_title('Self-transendence')
axes[3].set_xlabel('')
axes[3].set_ylabel('Dictionary based index')
axes[3].tick_params(axis='x', rotation=90)
ax32 = axes[3].twinx()

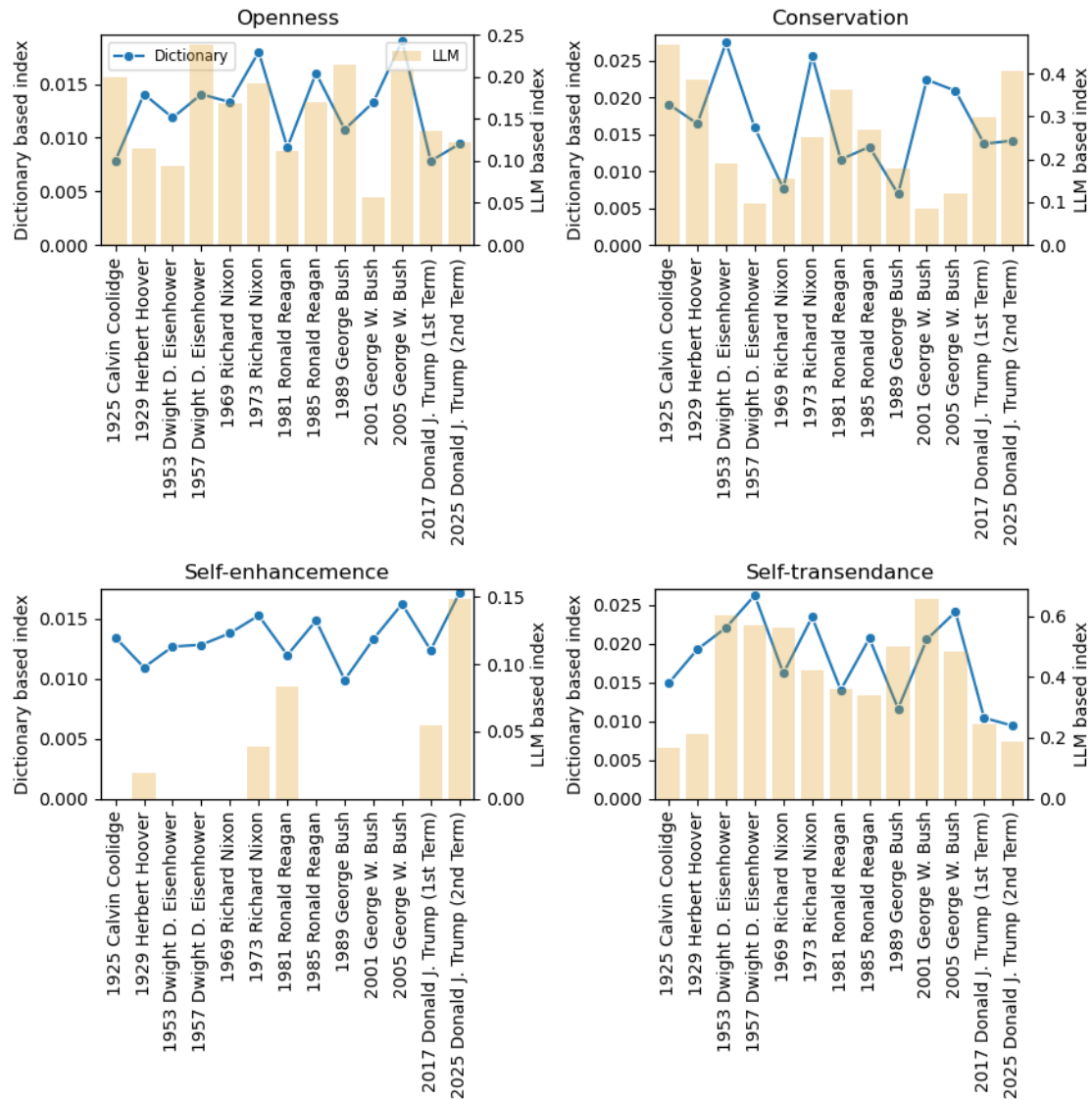
```

```

sns.barplot(ax=ax32, data=df_party, x='president_year',
            y='llm_selftranscendence', alpha=0.3, color='orange')
ax32.set_ylabel('LLM based index')
axes[3].set_ylim(bottom=0)

# Adjust layout
plt.tight_layout()
plt.show()

```



```

[275]: #examine self-enhancement text
df[(df['year'] == 1933) & (df['llm_value_mistral'] == 'self-enhancement')]

```

```

[275]:      doc_id      president  year  paragraph_idx  openness_count  \
387      8  Franklin D. Roosevelt  1933             26             1

      conservation_count  self_en_count  self_trancd_count  llm_value_mistral  \
387                    1              0              0  self-enhancement

      text  word_count  \
387  We do not distrust the future of essential dem...             60

      president_year      party  openness_dict  conservation_dict  \
387  1933 Franklin D. Roosevelt  Democrat  {'futur': 1}  {'disciplin': 1}

      self_en_dict  self_trancd_dict
387              {}              {}

```

```

[265]: df.loc[387, 'text' ]

```

```

[265]: 'We do not distrust the future of essential democracy. The people of the United
States have not failed. In their need they have registered a mandate that they
want direct, vigorous action. They have asked for discipline and direction under
leadership. They have made me the present instrument of their wishes. In the
spirit of the gift I take it.'

```

```

[274]: #examine openness text
df[(df['year'] == 2001)].sort_values('openness_count', ascending=False)

```

```

[274]:      doc_id      president  year  paragraph_idx  openness_count  \
765      52  George W. Bush  2001             30             3
750      52  George W. Bush  2001             15             3
743      52  George W. Bush  2001              8             2
761      52  George W. Bush  2001             26             2
752      52  George W. Bush  2001             17             2
738      52  George W. Bush  2001              3             1
748      52  George W. Bush  2001             13             1
766      52  George W. Bush  2001             31             1
762      52  George W. Bush  2001             27             1
751      52  George W. Bush  2001             16             1
755      52  George W. Bush  2001             20             1
745      52  George W. Bush  2001             10             1
740      52  George W. Bush  2001              5             1
741      52  George W. Bush  2001              6             1
747      52  George W. Bush  2001             12             1
749      52  George W. Bush  2001             14             0
742      52  George W. Bush  2001              7             0
771      52  George W. Bush  2001             36             0
770      52  George W. Bush  2001             35             0
769      52  George W. Bush  2001             34             0

```

768	52	George W. Bush	2001	33	0
767	52	George W. Bush	2001	32	0
764	52	George W. Bush	2001	29	0
763	52	George W. Bush	2001	28	0
744	52	George W. Bush	2001	9	0
760	52	George W. Bush	2001	25	0
759	52	George W. Bush	2001	24	0
758	52	George W. Bush	2001	23	0
757	52	George W. Bush	2001	22	0
756	52	George W. Bush	2001	21	0
739	52	George W. Bush	2001	4	0
754	52	George W. Bush	2001	19	0
753	52	George W. Bush	2001	18	0
746	52	George W. Bush	2001	11	0
772	52	George W. Bush	2001	37	0

	conservation_count	self_en_count	self_trandcd_count	llm_value_mistral \
765	0	1	0	self-transcendence
750	0	1	0	self-transcendence
743	2	0	2	openness
761	1	1	3	self-transcendence
752	2	2	1	conservation
738	3	1	3	neutral
748	1	1	1	self-transcendence
766	2	2	3	self-transcendence
762	2	1	3	self-transcendence
751	1	2	1	conservation
755	1	0	1	self-transcendence
745	1	3	1	self-transcendence
740	4	1	0	self-transcendence
741	1	0	0	self-transcendence
747	2	0	2	self-transcendence
749	0	1	3	self-transcendence
742	0	0	0	self-transcendence
771	0	0	0	neutral
770	0	0	0	neutral
769	0	0	0	neutral
768	0	0	1	neutral
767	1	0	0	self-transcendence
764	0	0	0	self-transcendence
763	0	1	0	self-transcendence
744	0	1	2	openness
760	0	1	0	self-transcendence
759	2	0	0	self-transcendence
758	2	0	1	self-transcendence
757	1	0	0	self-transcendence
756	3	1	4	self-transcendence

739	0	0	1	neutral
754	0	0	0	self-transcendence
753	3	1	0	conservation
746	2	0	1	self-transcendence
772	0	0	0	neutral

	text	word_count	\
765	Much time has passed since Jefferson arrived f...	36	
750	America at its best is also courageous. Our na...	65	
743	Through much of the last century, America's fa...	79	
761	I will live and lead by these principles: to a...	57	
752	We will build our defenses beyond challenge, l...	57	
738	Thank you, all. Chief Justice Rehnquist, Presi...	44	
748	Some seem to believe that our politics can aff...	84	
766	We are not this story's author, who fills time...	80	
762	What you do is as important as anything Govern...	59	
751	Together we will reclaim America's schools bef...	53	
755	Abandonment and abuse are not acts of God; the...	59	
745	Our unity, our Union, is a serious work of lea...	69	
740	I am honored and humbled to stand here where s...	95	
741	It is the American story, a story of flawed an...	46	
747	Today we affirm a new commitment to live out o...	48	
749	We must live up to the calling we share. Civil...	43	
742	Americans are called to enact this promise in ...	30	
771	TwitterFacebook	1	
770	The American Presidency ProjectJohn Woolley an...	8	
769	George W. Bush, Inaugural Address Online by Ge...	18	
768	NOTE: The President spoke at 12:05 p.m. at the...	28	
767	God bless you all, and God bless America.	8	
764	After the Declaration of Independence was sign...	44	
763	Americans are generous and strong and decent, ...	44	
744	While many of our citizens prosper, others dou...	64	
760	Sometimes in life we're called to do great thi...	41	
759	Our public interest depends on private charact...	28	
758	America at its best is a place where personal ...	65	
757	Many in our country do not know the pain of po...	47	
756	Government has great responsibilities for publ...	73	
739	As I begin, I thank President Clinton for his ...	29	
754	America at its best is compassionate. In the q...	42	
753	We will defend our allies and our interests. W...	40	
746	America has never been united by blood or birt...	62	
772	Copyright © The American Presidency ProjectTer...	8	

	president_year	party	openess_dict	\
765	2001 George W. Bush	Republican	{'chang': 1, 'dream': 1, 'courag': 1}	
750	2001 George W. Bush	Republican	{'futur': 1, 'courag': 2, 'inspir': 1}	
743	2001 George W. Bush	Republican	{'freedom': 1, 'hope': 1}	

761	2001	George W. Bush	Republican	{'advanc': 1, 'lead': 1}
752	2001	George W. Bush	Republican	{'liberti': 1, 'new': 2}
738	2001	George W. Bush	Republican	{'new': 1}
748	2001	George W. Bush	Republican	{'lead': 1}
766	2001	George W. Bush	Republican	{'renew': 1}
762	2001	George W. Bush	Republican	{'reform': 1}
751	2001	George W. Bush	Republican	{'reform': 1}
755	2001	George W. Bush	Republican	{'hope': 1}
745	2001	George W. Bush	Republican	{'lead': 1}
740	2001	George W. Bush	Republican	{'new': 1}
741	2001	George W. Bush	Republican	{'unfold': 1}
747	2001	George W. Bush	Republican	{'new': 1}
749	2001	George W. Bush	Republican	{}
742	2001	George W. Bush	Republican	{}
771	2001	George W. Bush	Republican	{}
770	2001	George W. Bush	Republican	{}
769	2001	George W. Bush	Republican	{}
768	2001	George W. Bush	Republican	{}
767	2001	George W. Bush	Republican	{}
764	2001	George W. Bush	Republican	{}
763	2001	George W. Bush	Republican	{}
744	2001	George W. Bush	Republican	{}
760	2001	George W. Bush	Republican	{}
759	2001	George W. Bush	Republican	{}
758	2001	George W. Bush	Republican	{}
757	2001	George W. Bush	Republican	{}
756	2001	George W. Bush	Republican	{}
739	2001	George W. Bush	Republican	{}
754	2001	George W. Bush	Republican	{}
753	2001	George W. Bush	Republican	{}
746	2001	George W. Bush	Republican	{}
772	2001	George W. Bush	Republican	{}

		conservation_dict	\
765			{}
750			{}
743			{'faith': 2}
761			{'respons': 1}
752			{'countri': 1, 'defens': 1}
738			{'tradit': 1}
748			{'countri': 1}
766			{'duti': 1, 'countri': 1}
762			{'respons': 1, 'defend': 1}
751			{'secur': 1}
755			{'order': 1}
745			{'principl': 1}
740			{'protect': 1, 'defend': 1, 'honor': 1, 'humbl...

```

741                                {'endur': 1}
747                    {'principl': 1, 'commit': 2}
749                                {}
742                                {}
771                                {}
770                                {}
769                                {}
768                                {}
767                                {'god': 2}
764                                {}
763                                {}
744                                {}
760                                {}
759                    {'famili': 1, 'duti': 1}
758                    {'respons': 2, 'commit': 1}
757                                {'countri': 1}
756                    {'respons': 1, 'honor': 1, 'safeti': 1}
739                                {}
754                                {}
753                    {'faith': 1, 'defend': 1}
746                    {'countri': 1, 'uphold': 1}
772                                {}

```

```

                                self_en_dict \
765                    {'courag': 1}
750                    {'courag': 2}
743                                {}
761                    {'lead': 1}
752                    {'power': 1, 'build': 1}
738                    {'author': 1}
748                    {'lead': 1}
766                    {'achiev': 1}
762                    {'build': 1}
751                    {'power': 1, 'secur': 1}
755                                {}
745                    {'power': 1, 'build': 1, 'lead': 1}
740                    {'power': 1}
741                                {}
747                                {}
749                    {'determin': 1}
742                                {}
771                                {}
770                                {}
769                                {}
768                                {}
767                                {}
764                                {}

```

```

763             {'strong': 1}
744             {'ambit': 1}
760             {'great': 2}
759             {}
758             {}
757             {}
756             {'great': 1}
739             {}
754             {}
753             {'resolv': 1}
746             {}
772             {}

                                self_trancd_dict
765                                     {}
750                                     {}
743             {'freedom': 1, 'hope': 1}
761             {'justic': 1, 'care': 1, 'serv': 1}
752                                     {'liberti': 1}
738             {'justic': 1, 'peac': 1, 'affirm': 1}
748                                     {'peac': 1}
766             {'digniti': 1, 'servic': 1, 'affirm': 1}
762             {'servic': 1, 'communiti': 1, 'serv': 1}
751                                     {'togeth': 1}
755                                     {'hope': 1}
745                                     {'justic': 1}
740                                     {}
741                                     {}
747             {'civil': 1, 'affirm': 1}
749             {'share': 1, 'civil': 1, 'communiti': 1}
742                                     {}
771                                     {}
770                                     {}
769                                     {}
768                                     {'justic': 1}
767                                     {}
764                                     {}
763                                     {}
744             {'justic': 1, 'share': 1}
760                                     {}
759                                     {}
758                                     {'communiti': 1}
757                                     {}
756 {'compass': 1, 'right': 1, 'civil': 1, 'commun...
739                                     {'servic': 1}
754                                     {}
753                                     {}

```



```

746                                     {'embrac': 1}
772                                     {}

```

```
[261]: df.loc[765, 'text' ]
```

```
[261]: "Much time has passed since Jefferson arrived for his inauguration. The years
and changes accumulate, but the themes of this day, he would know: our Nation's
grand story of courage and its simple dream of dignity."
```

```
[262]: df.loc[750, 'text' ]
```

```
[262]: 'America at its best is also courageous. Our national courage has been clear in
times of depression and war, when defeating common dangers defined our common
good. Now we must choose if the example of our fathers and mothers will inspire
us or condemn us. We must show courage in a time of blessing by confronting
problems instead of passing them on to future generations.'
```

```
[278]: #examine self transcendence values
df[(df['year'] == 2021)].sort_values('self_trancd_count', ascending=False)
```

```
[278]:
```

	doc_id	president	year	paragraph_idx	openess_count	\
785	53	Joseph R. Biden, Jr.	2021	12	1	
786	53	Joseph R. Biden, Jr.	2021	13	0	
787	53	Joseph R. Biden, Jr.	2021	14	0	
800	53	Joseph R. Biden, Jr.	2021	27	2	
797	53	Joseph R. Biden, Jr.	2021	24	1	
778	53	Joseph R. Biden, Jr.	2021	5	0	
781	53	Joseph R. Biden, Jr.	2021	8	1	
803	53	Joseph R. Biden, Jr.	2021	30	1	
814	53	Joseph R. Biden, Jr.	2021	41	2	
794	53	Joseph R. Biden, Jr.	2021	21	0	
792	53	Joseph R. Biden, Jr.	2021	19	0	
790	53	Joseph R. Biden, Jr.	2021	17	0	
789	53	Joseph R. Biden, Jr.	2021	16	1	
788	53	Joseph R. Biden, Jr.	2021	15	0	
812	53	Joseph R. Biden, Jr.	2021	39	1	
780	53	Joseph R. Biden, Jr.	2021	7	0	
817	53	Joseph R. Biden, Jr.	2021	44	0	
776	53	Joseph R. Biden, Jr.	2021	3	1	
810	53	Joseph R. Biden, Jr.	2021	37	0	
808	53	Joseph R. Biden, Jr.	2021	35	0	
809	53	Joseph R. Biden, Jr.	2021	36	1	
811	53	Joseph R. Biden, Jr.	2021	38	0	
816	53	Joseph R. Biden, Jr.	2021	43	0	
813	53	Joseph R. Biden, Jr.	2021	40	0	
815	53	Joseph R. Biden, Jr.	2021	42	0	
806	53	Joseph R. Biden, Jr.	2021	33	0	

818	53	Joseph R. Biden, Jr.	2021	45	0
819	53	Joseph R. Biden, Jr.	2021	46	0
820	53	Joseph R. Biden, Jr.	2021	47	0
807	53	Joseph R. Biden, Jr.	2021	34	0
799	53	Joseph R. Biden, Jr.	2021	26	0
805	53	Joseph R. Biden, Jr.	2021	32	0
804	53	Joseph R. Biden, Jr.	2021	31	0
802	53	Joseph R. Biden, Jr.	2021	29	0
801	53	Joseph R. Biden, Jr.	2021	28	0
777	53	Joseph R. Biden, Jr.	2021	4	0
798	53	Joseph R. Biden, Jr.	2021	25	0
796	53	Joseph R. Biden, Jr.	2021	23	0
795	53	Joseph R. Biden, Jr.	2021	22	0
793	53	Joseph R. Biden, Jr.	2021	20	0
791	53	Joseph R. Biden, Jr.	2021	18	0
784	53	Joseph R. Biden, Jr.	2021	11	0
783	53	Joseph R. Biden, Jr.	2021	10	1
782	53	Joseph R. Biden, Jr.	2021	9	1
779	53	Joseph R. Biden, Jr.	2021	6	0
821	53	Joseph R. Biden, Jr.	2021	48	0

	conservation_count	self_en_count	self_trancd_count	llm_value_mistral \
785	3	3	3	self-transcendence
786	1	1	3	self-transcendence
787	0	1	2	self-transcendence
800	0	0	2	self-transcendence
797	3	1	2	self-transcendence
778	0	2	2	self-transcendence
781	1	1	1	self-transcendence
803	0	3	1	openness
814	1	1	1	self-transcendence
794	0	0	1	openness
792	0	0	1	self-transcendence
790	2	0	1	openness
789	1	0	1	self-transcendence
788	0	0	1	self-transcendence
812	0	1	1	self-transcendence
780	0	2	1	self-transcendence
817	0	0	1	neutral
776	0	0	1	neutral
810	0	1	0	openness
808	1	0	0	self-transcendence
809	0	2	0	self-transcendence
811	0	0	0	self-transcendence
816	2	0	0	self-transcendence
813	3	1	0	self-transcendence
815	1	1	0	self-transcendence

806	0	0	0	neutral
818	0	0	0	self-transcendence
819	0	0	0	neutral
820	0	0	0	neutral
807	0	0	0	neutral
799	4	1	0	self-transcendence
805	0	0	0	self-transcendence
804	1	0	0	self-transcendence
802	1	1	0	self-transcendence
801	1	0	0	self-transcendence
777	0	1	0	openness
798	0	0	0	self-transcendence
796	1	1	0	self-transcendence
795	0	0	0	self-transcendence
793	0	1	0	self-transcendence
791	0	0	0	openness
784	0	0	0	self-transcendence
783	0	0	0	self-transcendence
782	2	1	0	self-transcendence
779	2	0	0	self-transcendence
821	0	0	0	neutral

	text	word_count	\
785	With unity we can do great things, important t...	72	
786	I know speaking of unity can sound to some lik...	77	
787	Through the Civil War, the Great Depression, W...	56	
800	Look, I understand that many of my fellow Amer...	164	
797	To all those who supported our campaign, I am ...	119	
778	So now, on this hallowed ground where just a f...	133	
781	Few people in our Nation's history have been m...	125	
803	Look, folks, all my colleagues that I served w...	104	
814	May this be the story that guides us, the stor...	76	
794	Here we stand, where 108 years ago, at another...	51	
792	Just look around. Here we stand, in the shadow...	37	
790	And so today, at this time, in this place, let...	72	
789	This is our historic moment of crisis and chal...	52	
788	We can see each other not as adversaries, but ...	54	
812	Let's add-let's us add our own work and prayer...	50	
780	This is a great Nation; we are good people. Ov...	69	
817	NOTE: The President spoke at 11:52 a.m. at the...	34	
776	Chief Justice Roberts, Vice President Harris, ...	59	
810	And when we do, we will write the next great c...	58	
808	Folks, this is a time of testing. We face an a...	71	
809	Now we're going to be tested. Are we going to ...	98	
811	"The work and prayers of centuries have brough...	41	
816	May God bless America, and may God protect our...	13	
813	My fellow Americans, I close today where I beg...	107	

815	So with purpose and resolve we turn to those t...	33
806	[At this point, a moment of silence was observ...	9
818	Joseph R. Biden, Jr., Inaugural Address Online...	19
819	The American Presidency ProjectJohn Woolley an...	8
820	TwitterFacebook	1
807	Amen.	1
799	Recent weeks and months have taught us a painf...	60
805	So I ask you, let's say a silent prayer for th...	25
804	Look, you all know, we have been through so mu...	77
802	My fellow Americans, in the work ahead of us, ...	81
801	Because here is the thing about life: There is...	72
777	Today we celebrate the triumph not of a candid...	54
798	Many centuries ago, Saint Augustine, a saint o...	60
796	And here we stand, just days after a riotous m...	51
795	Here we stand across the Potomac from Arlingto...	23
793	Here we stand looking out on the great Mall wh...	16
791	My fellow Americans, we have to be different t...	27
784	Today, on this January day, my whole soul is i...	49
783	In another January on New Year's Day in 1863, ...	54
782	To overcome these challenges-to restore the so...	31
779	I have just taken the sacred oath each of the...	51
821	Copyright © The American Presidency ProjectTer...	8

	president_year	party	openness_dict \
785	2021	Joseph R. Biden, Jr. Democrat	{ 'lead': 1 }
786	2021	Joseph R. Biden, Jr. Democrat	{ }
787	2021	Joseph R. Biden, Jr. Democrat	{ }
800	2021	Joseph R. Biden, Jr. Democrat	{ 'futur': 1, 'open': 1 }
797	2021	Joseph R. Biden, Jr. Democrat	{ 'lead': 1 }
778	2021	Joseph R. Biden, Jr. Democrat	{ }
781	2021	Joseph R. Biden, Jr. Democrat	{ 'dream': 1 }
803	2021	Joseph R. Biden, Jr. Democrat	{ 'lead': 1 }
814	2021	Joseph R. Biden, Jr. Democrat	{ 'liberti': 1, 'inspir': 1 }
794	2021	Joseph R. Biden, Jr. Democrat	{ }
792	2021	Joseph R. Biden, Jr. Democrat	{ }
790	2021	Joseph R. Biden, Jr. Democrat	{ }
789	2021	Joseph R. Biden, Jr. Democrat	{ 'path': 1 }
788	2021	Joseph R. Biden, Jr. Democrat	{ }
812	2021	Joseph R. Biden, Jr. Democrat	{ 'unfold': 1 }
780	2021	Joseph R. Biden, Jr. Democrat	{ }
817	2021	Joseph R. Biden, Jr. Democrat	{ }
776	2021	Joseph R. Biden, Jr. Democrat	{ 'renew': 1 }
810	2021	Joseph R. Biden, Jr. Democrat	{ }
808	2021	Joseph R. Biden, Jr. Democrat	{ }
809	2021	Joseph R. Biden, Jr. Democrat	{ 'new': 1 }
811	2021	Joseph R. Biden, Jr. Democrat	{ }
816	2021	Joseph R. Biden, Jr. Democrat	{ }

813	2021	Joseph R. Biden, Jr.	Democrat	{}
815	2021	Joseph R. Biden, Jr.	Democrat	{}
806	2021	Joseph R. Biden, Jr.	Democrat	{}
818	2021	Joseph R. Biden, Jr.	Democrat	{}
819	2021	Joseph R. Biden, Jr.	Democrat	{}
820	2021	Joseph R. Biden, Jr.	Democrat	{}
807	2021	Joseph R. Biden, Jr.	Democrat	{}
799	2021	Joseph R. Biden, Jr.	Democrat	{}
805	2021	Joseph R. Biden, Jr.	Democrat	{}
804	2021	Joseph R. Biden, Jr.	Democrat	{}
802	2021	Joseph R. Biden, Jr.	Democrat	{}
801	2021	Joseph R. Biden, Jr.	Democrat	{}
777	2021	Joseph R. Biden, Jr.	Democrat	{}
798	2021	Joseph R. Biden, Jr.	Democrat	{}
796	2021	Joseph R. Biden, Jr.	Democrat	{}
795	2021	Joseph R. Biden, Jr.	Democrat	{}
793	2021	Joseph R. Biden, Jr.	Democrat	{}
791	2021	Joseph R. Biden, Jr.	Democrat	{}
784	2021	Joseph R. Biden, Jr.	Democrat	{}
783	2021	Joseph R. Biden, Jr.	Democrat	{'new': 1}
782	2021	Joseph R. Biden, Jr.	Democrat	{'futur': 1}
779	2021	Joseph R. Biden, Jr.	Democrat	{}
821	2021	Joseph R. Biden, Jr.	Democrat	{}

		conservation_dict \	
785		{'secur': 1, 'uniti': 1, 'safe': 1}	
786		{'uniti': 1}	
787		{}	
800		{}	
797		{'faith': 1, 'humbl': 1}	
778		{}	
781		{'countri': 1}	
803		{}	
814		{'secur': 1}	
794		{}	
792		{}	
790		{'cultur': 1, 'respect': 1}	
789		{'uniti': 1}	
788		{}	
812		{}	
780		{}	
817		{}	
776		{}	
810		{}	
808		{'respons': 1}	
809		{}	
811		{}	

```

816             {'god': 2, 'protect': 1}
813         {'god': 1, 'defend': 3, 'sacr': 1}
815             {'countri': 1}
806             {}
818             {}
819             {}
820             {}
807             {}
799 {'duti': 1, 'protect': 1, 'defend': 1, 'honor'...
805             {}
804             {'honor': 1}
802             {'endur': 1}
801             {'countri': 1}
777             {}
798             {}
796             {'sacr': 1}
795             {}
793             {}
791             {}
784             {}
783             {}
782             {'secur': 1, 'restor': 1}
779             {'patriot': 1, 'sacr': 1}
821             {}

```

	self_en_dict	self_trancd_dict
785	{'great': 1, 'secur': 1, 'lead': 1}	{'care': 1, 'uniti': 1, 'right': 1}
786	{'victori': 1}	{'equal': 1, 'uniti': 1}
787	{'great': 1}	{'togeth': 1, 'civil': 1}
800	{}	{'toler': 1, 'open': 1}
797	{'lead': 1}	{'right': 1, 'support': 3}
778	{'power': 1, 'strength': 1}	{'peac': 1, 'togeth': 1}
781	{'rise': 1}	{'justic': 2}
803	{'power': 1, 'strong': 1, 'lead': 1}	{'serv': 1}
814	{'secur': 1}	{'liberti': 1}
794	{}	{'right': 1}
792	{}	{'civil': 1}
790	{}	{'respect': 1}
789	{}	{'uniti': 1}
788	{}	{'digniti': 1}
812	{'great': 1}	{'heal': 1}
780	{'great': 1, 'forward': 1}	{'peac': 1}
817	{}	{'justic': 1}
776	{}	{'justic': 1}
810	{'great': 1}	{}
808	{}	{}
809	{'resolv': 1, 'rise': 1}	{}

```

811                                {}                                {}
816                                {}                                {}
813                                {'great': 1}                    {}
815                                {'resolv': 1}                    {}
806                                {}                                {}
818                                {}                                {}
819                                {}                                {}
820                                {}                                {}
807                                {}                                {}
799                                {'power': 1}                    {}
805                                {}                                {}
804                                {}                                {}
802                                {'strength': 1}                  {}
801                                {}                                {}
777                                {'triumph': 1}                  {}
798                                {}                                {}
796                                {'drive': 1}                    {}
795                                {}                                {}
793                                {'great': 1}                    {}
791                                {}                                {}
784                                {}                                {}
783                                {}                                {}
782                                {'secur': 1}                    {}
779                                {}                                {}
821                                {}                                {}

```

```

[279]: df[(df['year'] == 2021) & (df['llm_value_mistral'] == 'self-transcendence') &
        (df['self_trancd_count'] == 0)]

```

```

[279]:   doc_id  president  year  paragraph_idx  openness_count  \
779     53  Joseph R. Biden, Jr.  2021           6             0
782     53  Joseph R. Biden, Jr.  2021           9             1
783     53  Joseph R. Biden, Jr.  2021          10             1
784     53  Joseph R. Biden, Jr.  2021          11             0
793     53  Joseph R. Biden, Jr.  2021          20             0
795     53  Joseph R. Biden, Jr.  2021          22             0
796     53  Joseph R. Biden, Jr.  2021          23             0
798     53  Joseph R. Biden, Jr.  2021          25             0
799     53  Joseph R. Biden, Jr.  2021          26             0
801     53  Joseph R. Biden, Jr.  2021          28             0
802     53  Joseph R. Biden, Jr.  2021          29             0
804     53  Joseph R. Biden, Jr.  2021          31             0
805     53  Joseph R. Biden, Jr.  2021          32             0
808     53  Joseph R. Biden, Jr.  2021          35             0
809     53  Joseph R. Biden, Jr.  2021          36             1
811     53  Joseph R. Biden, Jr.  2021          38             0
813     53  Joseph R. Biden, Jr.  2021          40             0

```

815	53	Joseph R. Biden, Jr.	2021	42	0
816	53	Joseph R. Biden, Jr.	2021	43	0
818	53	Joseph R. Biden, Jr.	2021	45	0

	conservation_count	self_en_count	self_trancd_count	llm_value_mistral \
779	2	0	0	self-transcendence
782	2	1	0	self-transcendence
783	0	0	0	self-transcendence
784	0	0	0	self-transcendence
793	0	1	0	self-transcendence
795	0	0	0	self-transcendence
796	1	1	0	self-transcendence
798	0	0	0	self-transcendence
799	4	1	0	self-transcendence
801	1	0	0	self-transcendence
802	1	1	0	self-transcendence
804	1	0	0	self-transcendence
805	0	0	0	self-transcendence
808	1	0	0	self-transcendence
809	0	2	0	self-transcendence
811	0	0	0	self-transcendence
813	3	1	0	self-transcendence
815	1	1	0	self-transcendence
816	2	0	0	self-transcendence
818	0	0	0	self-transcendence

	text	word_count \
779	I have just taken the sacred oath each of thes...	51
782	To overcome these challenges-to restore the so...	31
783	In another January on New Year's Day in 1863, ...	54
784	Today, on this January day, my whole soul is i...	49
793	Here we stand looking out on the great Mall wh...	16
795	Here we stand across the Potomac from Arlingto...	23
796	And here we stand, just days after a riotous m...	51
798	Many centuries ago, Saint Augustine, a saint o...	60
799	Recent weeks and months have taught us a painf...	60
801	Because here is the thing about life: There is...	72
802	My fellow Americans, in the work ahead of us, ...	81
804	Look, you all know, we have been through so mu...	77
805	So I ask you, let's say a silent prayer for th...	25
808	Folks, this is a time of testing. We face an a...	71
809	Now we're going to be tested. Are we going to ...	98
811	"The work and prayers of centuries have brough...	41
813	My fellow Americans, I close today where I beg...	107
815	So with purpose and resolve we turn to those t...	33
816	May God bless America, and may God protect our...	13
818	Joseph R. Biden, Jr., Inaugural Address Online...	19

	president_year	party	openess_dict	\
779	2021	Joseph R. Biden, Jr.	Democrat	{}
782	2021	Joseph R. Biden, Jr.	Democrat	{'futur': 1}
783	2021	Joseph R. Biden, Jr.	Democrat	{'new': 1}
784	2021	Joseph R. Biden, Jr.	Democrat	{}
793	2021	Joseph R. Biden, Jr.	Democrat	{}
795	2021	Joseph R. Biden, Jr.	Democrat	{}
796	2021	Joseph R. Biden, Jr.	Democrat	{}
798	2021	Joseph R. Biden, Jr.	Democrat	{}
799	2021	Joseph R. Biden, Jr.	Democrat	{}
801	2021	Joseph R. Biden, Jr.	Democrat	{}
802	2021	Joseph R. Biden, Jr.	Democrat	{}
804	2021	Joseph R. Biden, Jr.	Democrat	{}
805	2021	Joseph R. Biden, Jr.	Democrat	{}
808	2021	Joseph R. Biden, Jr.	Democrat	{}
809	2021	Joseph R. Biden, Jr.	Democrat	{'new': 1}
811	2021	Joseph R. Biden, Jr.	Democrat	{}
813	2021	Joseph R. Biden, Jr.	Democrat	{}
815	2021	Joseph R. Biden, Jr.	Democrat	{}
816	2021	Joseph R. Biden, Jr.	Democrat	{}
818	2021	Joseph R. Biden, Jr.	Democrat	{}

	conservation_dict	\
779	{'patriot': 1, 'sacr': 1}	
782	{'secur': 1, 'restor': 1}	
783		{}
784		{}
793		{}
795		{}
796	{'sacr': 1}	
798		{}
799	{'duti': 1, 'protect': 1, 'defend': 1, 'honor'...	
801	{'countri': 1}	
802	{'endur': 1}	
804	{'honor': 1}	
805		{}
808	{'respons': 1}	
809		{}
811		{}
813	{'god': 1, 'defend': 3, 'sacr': 1}	
815	{'countri': 1}	
816	{'god': 2, 'protect': 1}	
818		{}

	self_en_dict	self_trancd_dict
779	{}	{}

```

782         {'secur': 1}          {}
783             {}              {}
784             {}              {}
793         {'great': 1}         {}
795             {}              {}
796         {'drive': 1}          {}
798             {}              {}
799         {'power': 1}          {}
801             {}              {}
802         {'strength': 1}       {}
804             {}              {}
805             {}              {}
808             {}              {}
809     {'resolv': 1, 'rise': 1}   {}
811             {}              {}
813         {'great': 1}         {}
815         {'resolv': 1}         {}
816             {}              {}
818             {}              {}

```

```
[282]: df.loc[783, 'text' ]
```

```
[282]: 'In another January on New Year\'s Day in 1863, Abraham Lincoln signed the
Emancipation Proclamation. When he put pen to paper, the President said, and I
quote, "If my name ever goes down into history it will be for this act, and my
whole soul is in it." My whole soul is in it.'
```

```
[285]: df.loc[795, 'text' ]
```

```
[285]: 'Here we stand across the Potomac from Arlington Cemetery, where heroes who gave
the last full measure of devotion rest in eternal peace.'
```

```
[302]: # plot llm index as percentage for easy looking trend
# Step 1: Group and count
grouped = df.groupby(['party', 'president_year', 'llm_value_mistral']).size().
    ↪reset_index(name='count')

# Step 2: Pivot for plotting
pivoted = grouped.pivot_table(index=['party', 'president_year'],
                                columns='llm_value_mistral',
                                values='count',
                                fill_value=0)

# Step 3: Convert counts to percentages (row-wise)
percentages = pivoted.div(pivoted.sum(axis=1), axis=0) * 100

# Step 4: Split by party

```

```

democrat_data = percentages.loc['Democrat']
republican_data = percentages.loc['Republican']

# Step 5: Plotting
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 6), sharex=False)

# Democrat subplot
democrat_data.plot(kind='bar', stacked=True, ax=axes[0],
    colormap='Set2', legend=False)
axes[0].set_title('Democrat Presidents - LLM Values')
axes[0].set_ylabel('Percentage')
axes[0].set_xlabel('')

# Republican subplot
republican_data.plot(kind='bar', stacked=True, ax=axes[1],
    colormap='Set2', legend=False)
axes[1].set_title('Republican Presidents - LLM Values')
axes[1].set_ylabel('Percentage')
axes[1].set_xlabel('')

# Final touches
plt.xlabel('')
plt.tight_layout()
plt.show()

```

