Part 3 - Themes Over Time and Party Affiliation

Navein Clara Calvin Halasya

Tuesday 16th December, 2025

- What are the most common themes in inaugural addresses?
- How has this changed over time?
- Does a certain party have more polarizing speech?

```
import pandas as pd
import numpy as np
import spacy
from collections import Counter
```

```
inaugural = pd.read_csv('data/inaugural_address.csv').iloc[:, 1:]
#convert to dt
inaugural['date'] = pd.to_datetime(inaugural['date'])
inaugural.head()
```

	president_name	president_number	date	text
0	George Washington	1	1789-04-30 00:00:00+00:00	\nFellow-Citizens of the Senate and of the Hou...
1	George Washington	1	1793-03-04 00:00:00+00:00	\nFellow Citizens:\nI AM again called upon by ...
2	John Adams	2	1797-03-04 00:00:00+00:00	\nWHEN it was first perceived, in early times,...
3	Thomas Jefferson	3	1801-03-04 00:00:00+00:00	\nFriends and Fellow-Citizens:\nCALLED upon to...
4	Thomas Jefferson	3	1805-03-04 00:00:00+00:00	\nPROCEEDING, fellow-citizens, to that qualifi...

1 What are the most common themes in inaugural addresses?

```
from gensim.corpora import Dictionary
from gensim.models import LdaModel
import pyLDAvis
import pyLDAvis.gensim_models
from spacy import displacy

# If did not get env from Makefile, run:
!python -m spacy download en_core_web_sm

nlp = spacy.load("en_core_web_sm")

def preprocess_text(text):
    doc = nlp(text)
    return [
        token.lemma_.lower() for token in doc
        if not (token.is_stop or token.is_punct or token.is_space)
        and len(token.lemma_) > 3
    ]

processed_docs = inaugural["text"].apply(preprocess_text)

dictionary = Dictionary(processed_docs)
dictionary.filter_extremes(no_below=5, no_above=0.5)
corpus = [dictionary.doc2bow(doc) for doc in processed_docs] # approx 25 seconds

lda_model = LdaModel(corpus=corpus, id2word=dictionary, num_topics=5, random_state=42, passe

print("Inaugural Addresses LDA Themes:")
for i, topic in lda_model.print_topics( -1):
    print(f"Theme: {i}")
    print(f"Words: {topic}")
    print()
```

```
Inaugural Addresses LDA Themes:
Theme: 0
Words: 0.005*"method" + 0.004*"establish" + 0.004*"race" + 0.004*"community" + 0.004*"rest"

Theme: 1
Words: 0.010*"business" + 0.006*"federal" + 0.006*"increase" + 0.005*"ought" + 0.005*"legis

Theme: 2
Words: 0.009*"thank" + 0.008*"today" + 0.007*"like" + 0.006*"task" + 0.006*"face" + 0.005*"

Theme: 3
Words: 0.013*"today" + 0.012*"americans" + 0.009*"century" + 0.007*"democracy" + 0.007*"chi

Theme: 4
Words: 0.005*"opinion" + 0.005*"object" + 0.004*"general" + 0.004*"exist" + 0.004*"revenue"
```

```
pyLDAvis.enable_notebook()
pyLDAvis.gensim_models.prepare(lda_model, corpus, dictionary)
```

2 How have the major themes changed over time?

```
inaugural_1700s = inaugural[inaugural['date'].dt.year < 1800]
inaugural_1800s = inaugural[(1800 <= inaugural['date'].dt.year) & (inaugural['date'].dt.yea
inaugural_1900s = inaugural[(1900 <= inaugural['date'].dt.year) & (inaugural['date'].dt.yea
inaugural_2000s = inaugural[2000 < inaugural['date'].dt.year]

processed_docs_1700s = inaugural_1700s["text"].apply(preprocess_text)
processed_docs_1800s = inaugural_1800s["text"].apply(preprocess_text)
processed_docs_1900s = inaugural_1900s["text"].apply(preprocess_text)
```

```
processed_docs_2000s = inaugural_2000s["text"].apply(preprocess_text)

# use same dictionary for all:
dictionary = Dictionary(processed_docs)
dictionary.filter_extremes(no_below=5, no_above=0.5)

corpus_1700s = [dictionary.doc2bow(doc) for doc in processed_docs_1700s]
corpus_1800s = [dictionary.doc2bow(doc) for doc in processed_docs_1800s]
corpus_1900s = [dictionary.doc2bow(doc) for doc in processed_docs_1900s]
corpus_2000s = [dictionary.doc2bow(doc) for doc in processed_docs_2000s]

lda_model_1700s = LdaModel(corpus=corpus_1700s, id2word=dictionary, num_topics=5, random_sta
lda_model_1800s = LdaModel(corpus=corpus_1800s, id2word=dictionary, num_topics=5, random_sta
lda_model_1900s = LdaModel(corpus=corpus_1900s, id2word=dictionary, num_topics=5, random_sta
lda_model_2000s = LdaModel(corpus=corpus_2000s, id2word=dictionary, num_topics=5, random_sta

print("Inaugural Addresses 18th Century LDA Themes:")
for i, topic in lda_model_1700s.print_topics( -1):
    print(f"Theme: {i}")
    print(f"Words: {topic}")
    print()
```

```
Inaugural Addresses 18th Century LDA Themes:
Theme: 0
Words: 0.003*"voice" + 0.003*"execute" + 0.003*"official" + 0.003*"entertain" + 0.003*"insta

Theme: 1
Words: 0.001*"knowledge" + 0.001*"general" + 0.001*"legislature" + 0.001*"choice" + 0.001*"e

Theme: 2
Words: 0.001*"happiness" + 0.001*"ought" + 0.001*"establish" + 0.001*"blessing" + 0.001*"hum

Theme: 3
Words: 0.007*"ought" + 0.005*"happiness" + 0.005*"establish" + 0.005*"circumstance" + 0.005*

Theme: 4
Words: 0.008*"legislature" + 0.006*"virtuous" + 0.006*"general" + 0.006*"choice" + 0.006*"kn
```

```
print("Inaugural Addresses 19th Century LDA Themes:")
for i, topic in lda_model_1800s.print_topics( -1):
    print(f"Theme: {i}")
    print(f"Words: {topic}")
    print()
```

```
Inaugural Addresses 19th Century LDA Themes:
Theme: 0
Words: 0.006*"protection" + 0.005*"object" + 0.005*"revenue" + 0.005*"extend" + 0.004*"impor
```

```
Theme: 1
Words: 0.008*"revenue" + 0.007*"business" + 0.006*"legislation" + 0.005*"countryman" + 0.005

Theme: 2
Words: 0.006*"officer" + 0.006*"revenue" + 0.005*"community" + 0.005*"method" + 0.005*"incre

Theme: 3
Words: 0.006*"opinion" + 0.005*"experience" + 0.005*"happiness" + 0.004*"feel" + 0.004*"circ

Theme: 4
Words: 0.005*"object" + 0.005*"general" + 0.005*"effect" + 0.005*"exist" + 0.005*"case" + 0.
```

```
print("Inaugural Addresses 20th Century LDA Themes:")
for i, topic in lda_model_1900s.print_topics( -1):
    print(f"Theme: {i}")
    print(f"Words: {topic}")
    print()
```

```
Inaugural Addresses 20th Century LDA Themes:
Theme: 0
Words: 0.007*"island" + 0.007*"rest" + 0.005*"inhabitant" + 0.005*"million" + 0.005*"faithfu

Theme: 1
Words: 0.010*"business" + 0.007*"increase" + 0.006*"ought" + 0.006*"race" + 0.005*"proper" +

Theme: 2
Words: 0.008*"task" + 0.006*"civilization" + 0.006*"opportunity" + 0.006*"face" + 0.005*"tho

Theme: 3
Words: 0.014*"today" + 0.012*"century" + 0.012*"americans" + 0.008*"help" + 0.007*"earth" +

Theme: 4
Words: 0.001*"today" + 0.001*"federal" + 0.001*"economic" + 0.001*"business" + 0.001*"opport
```

```
print("Inaugural Addresses 21st Century LDA Themes:")
for i, topic in lda_model_2000s.print_topics( -1):
    print(f"Theme: {i}")
    print(f"Words: {topic}")
    print()
```

```
Inaugural Addresses 21st Century LDA Themes:
Theme: 0
Words: 0.001*"thank" + 0.001*"today" + 0.001*"generation" + 0.001*"child" + 0.001*"americans

Theme: 1
Words: 0.006*"permanent" + 0.006*"excuse" + 0.006*"feel" + 0.005*"tyranny" + 0.005*"deep" +
Theme: 2
Words: 0.018*"thank" + 0.014*"today" + 0.014*"americans" + 0.009*"child" + 0.009*"democracy"

Theme: 3
Words: 0.011*"today" + 0.010*"generation" + 0.009*"americans" + 0.007*"woman" + 0.007*"chil

Theme: 4
Words: 0.001*"today" + 0.001*"child" + 0.001*"generation" + 0.001*"thank" + 0.001*"americans
```

```
# 18th century topics visaualization
pyLDAvis.gensim_models.prepare(lda_model_1700s, corpus_1700s, dictionary)

# 19th century topics visaualization
pyLDAvis.gensim_models.prepare(lda_model_1800s, corpus_1800s, dictionary)

# 20th century topics visaualization
pyLDAvis.gensim_models.prepare(lda_model_1900s, corpus_1900s, dictionary)

# 21st century topics visaualization
pyLDAvis.gensim_models.prepare(lda_model_2000s, corpus_2000s, dictionary)
```

3 Does a certain party have more polarizing speech?

```
inaugural['party'] = np.array([
    'None',
    'None',
    'Federalist',
    'Democratic -Republican',
    'Democratic -Republican',
    'Democratic -Republican',
    'Democratic -Republican',
    'Democratic -Republican',
    'Democratic -Republican',
    'National Republican',
    'Democratic',
    'Democratic',
    'Democratic',
    'Whig',
    'Democratic',
    'Whig',
    'Democratic',
    'Democratic',
    'Republican',
    'Republican',
    'Republican',
    'Republican',
    'Republican',
    'Republican',
    'Democratic',
    'Republican',
    'Democratic',
    'Republican',
    'Republican',
    'Republican',
    'Republican',
    'Democratic',
    'Democratic',
    'Republican',
    'Republican',
    'Republican',
    'Democratic',
    'Democratic',
    'Democratic',
    'Democratic',
    'Republican',
    'Democratic',
    'Republican',
    'Republican',
    'Republican',
    'Republican',
    'Democratic',
    'Democratic',
    'Republican',
    'Republican',
    'Democratic',
    'Democratic',
    'Republican',
    'Democratic',
    'Republican'
])
inaugural = inaugural[['president_name', 'party', 'president_number', 'date', 'text']]
inaugural.head()
```

	president_name	party	president_number	date	text
0	George Washington	None	1	1789-04-30 00:00:00+00:00	\nFellow-Citizens of the Senate and of the Hou...
1	George Washington	None	1	1793-03-04 00:00:00+00:00	\nFellow Citizens:\nI AM again called upon by ...
2	John Adams	Federalist	2	1797-03-04 00:00:00+00:00	\nWHEN it was first perceived, in early times,...
3	Thomas Jefferson	Democratic-Republican	3	1801-03-04 00:00:00+00:00	\nFriends and Fellow-Citizens:\nCALLED upon to...
4	Thomas Jefferson	Democratic-Republican	3	1805-03-04 00:00:00+00:00	\nPROCEEDING, fellow-citizens, to that qualifi...

```
from afinn import Afinn
import warnings
# ignore groupby warnings:
warnings.filterwarnings("ignore")

afinn = Afinn()

def calculate_sentiment_metrics(tokens):
    """
    Calculates sentiment metrics for tokens
    """
    scores = [afinn.score(token) for token in tokens]

    # Remove neutral words for polarization_score
    non_zero_scores = [s for s in scores if s != 0]
    return {
        'sentiment_score': sum(scores),
        'positive_words': sum(1 for s in scores if s > 0),
        'negative_words': sum(1 for s in scores if s < 0),
        'sentiment_variance': np.var(non_zero_scores) if non_zero_scores else 0
    }

sentiment_metrics = processed_docs.apply(calculate_sentiment_metrics)
sentiment_df = pd.DataFrame(sentiment_metrics.tolist())

# Combine with original data
inaugural_with_sentiment = pd.concat([inaugural, sentiment_df], axis=1)

def calculate_polarization(group):
    """
    Polarization metrics function for groupby
    """
    return pd.Series({
        'mean_sentiment_variance': group['sentiment_variance'].mean(),
        'total_positive_words': group['positive_words'].sum(),
        'total_negative_words': group['negative_words'].sum(),
        'net_sentiment': group['sentiment_score'].mean(),
        'num_speeches': len(group)
    })

polarization_by_party = inaugural_with_sentiment.groupby('party').apply(calculate_polarizati

polarization_by_party['avg_positive_per_speech'] = (
    polarization_by_party['total_positive_words'] / polarization_by_party['num_speeches']
)
polarization_by_party['avg_negative_per_speech'] = (
    polarization_by_party['total_negative_words'] / polarization_by_party['num_speeches']
)
```

```
/Users/calvin/miniconda3/envs/inaugural -address/lib/python3.11/site -packages/afinn/afinn.p
  self._word_pattern = re.compile('\w+', flags=re.UNICODE)
```

```
(
    polarization_by_party
    .sort_values('net_sentiment', ascending=False)
    [['total_positive_words', 'total_negative_words', 'net_sentiment', 'num_speeches']]
)
```

	total_positive_words	total_negative_words	net_sentiment	num_speeches
party				
Whig	478.0	180.0	245.000000	2.0
National Republican	161.0	45.0	227.000000	1.0
Federalist	147.0	40.0	203.000000	1.0
Democratic-Republican	887.0	288.0	194.000000	6.0
Republican	3718.0	1400.0	189.086957	23.0
Democratic	2638.0	1016.0	151.050000	20.0
None	92.0	23.0	69.000000	2.0

```
(
    polarization_by_party
    .sort_values('avg_positive_per_speech', ascending=False)
    [['avg_positive_per_speech', 'avg_negative_per_speech', 'mean_sentiment_variance']]
)
```

	avg_positive_per_speech	avg_negative_per_speech	mean_sentiment_variance
party			
Whig	239.000000	90.000000	2.588256
Republican	161.652174	60.869565	3.243731
National Republican	161.000000	45.000000	2.751744
Democratic-Republican	147.833333	48.000000	3.147726
Federalist	147.000000	40.000000	3.329578
Democratic	131.900000	50.800000	3.158040
None	46.000000	11.500000	2.705604