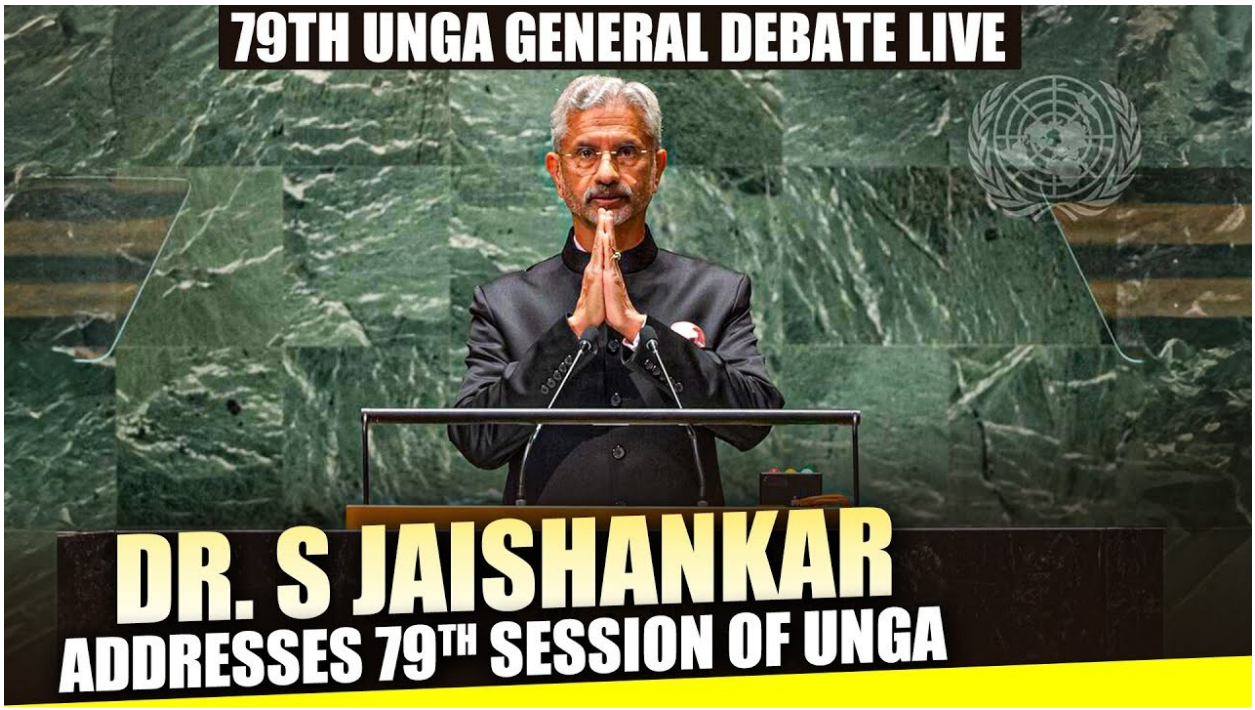
Exploratory Data Analysis, Named Entity Recognition, and Data Visualization on the National Statement by External Affairs Minister, Dr. S. Jaishankar at the General Debate of the 79th UN General Assembly




```
import pandas as pd

data = {
    "Section": ["Introduction", "Global Challenges", "Call for
Multilateral Reform",
                "India's Domestic Initiatives", "India's Global
Contributions",
                "Digital Transformation", "Globalization and Economic
Fairness",
                "Peace, Development and Conflict Resolution",
"International Law and Global Security",
                "Terrorism and Pakistan", "UN Reform and
Inclusiveness", "Closing Remarks"],

    "Details": [
```

```
        "Greetings from 1.4 billion people of Bharat; congratulated the UNGA President, supported the theme 'Leaving no one behind'.",
        "World recovering from Covid, war in Ukraine, conflict in Gaza; development plans disrupted in Global South, unfair trade practices, and rising debt.",
        "Trust has eroded, processes broken down; international system being exploited; need for reforming multilateralism to ensure peace and sustainable development.",
        "India focuses on vulnerable populations (women, youth, farmers), providing access to essential services (water, electricity, housing); targeted policies and financial support for employment and entrepreneurship.",
        "India's contributions include digital public infrastructure, supporting 78 nations, maritime security, humanitarian responses, and convening Global South Summits.",
        "India highlights the transformative potential of digital solutions, using fintech and technology for empowerment, not domination, to ensure no one is left behind.",
        "Criticism of unfair globalization model; call to democratize global production, build resilient supply chains, and ensure digital service access for all.",
        "Emphasis on resolving conflicts like Ukraine and Gaza for peace and development; call for the international community to find urgent solutions.",
        "Respect for international law is essential for global security; those leading must set examples, terrorism must be opposed in all forms.",
        "Condemnation of Pakistan's cross-border terrorism and radicalization; clear stance that Pakistan's misdeeds will have consequences, calls for Pakistan to vacate illegally occupied Indian territory.",
        "UN must be reformed to be more representative and inclusive; a central platform for addressing contemporary global concerns is necessary.",
        "Closing remarks with optimism for collective action, urging UN members to work together for a better world."
    ]
}

df = pd.DataFrame(data)

df
```

	Section	\
0	Introduction	
1	Global Challenges	
2	Call for Multilateral Reform	
3	India's Domestic Initiatives	
4	India's Global Contributions	
5	Digital Transformation	

```
6          Globalization and Economic Fairness
7   Peace, Development and Conflict Resolution
8        International Law and Global Security
9                        Terrorism and Pakistan
10                UN Reform and Inclusiveness
11                             Closing Remarks
```

```
                                                 Details
0   Greetings from 1.4 billion people of Bharat; c...
1   World recovering from Covid, war in Ukraine, c...
2   Trust has eroded, processes broken down; inter...
3   India focuses on vulnerable populations (women...
4   India's contributions include digital public i...
5   India highlights the transformative potential ...
6   Criticism of unfair globalization model; call ...
7   Emphasis on resolving conflicts like Ukraine a...
8   Respect for international law is essential for...
9   Condemnation of Pakistan's cross-border terror...
10  UN must be reformed to be more representative ...
11  Closing remarks with optimism for collective a...
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from collections import Counter
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.sentiment import SentimentIntensityAnalyzer
from sklearn.feature_extraction.text import CountVectorizer
from textblob import TextBlob
```

```
nltk.download('stopwords')
nltk.download('punkt')
```

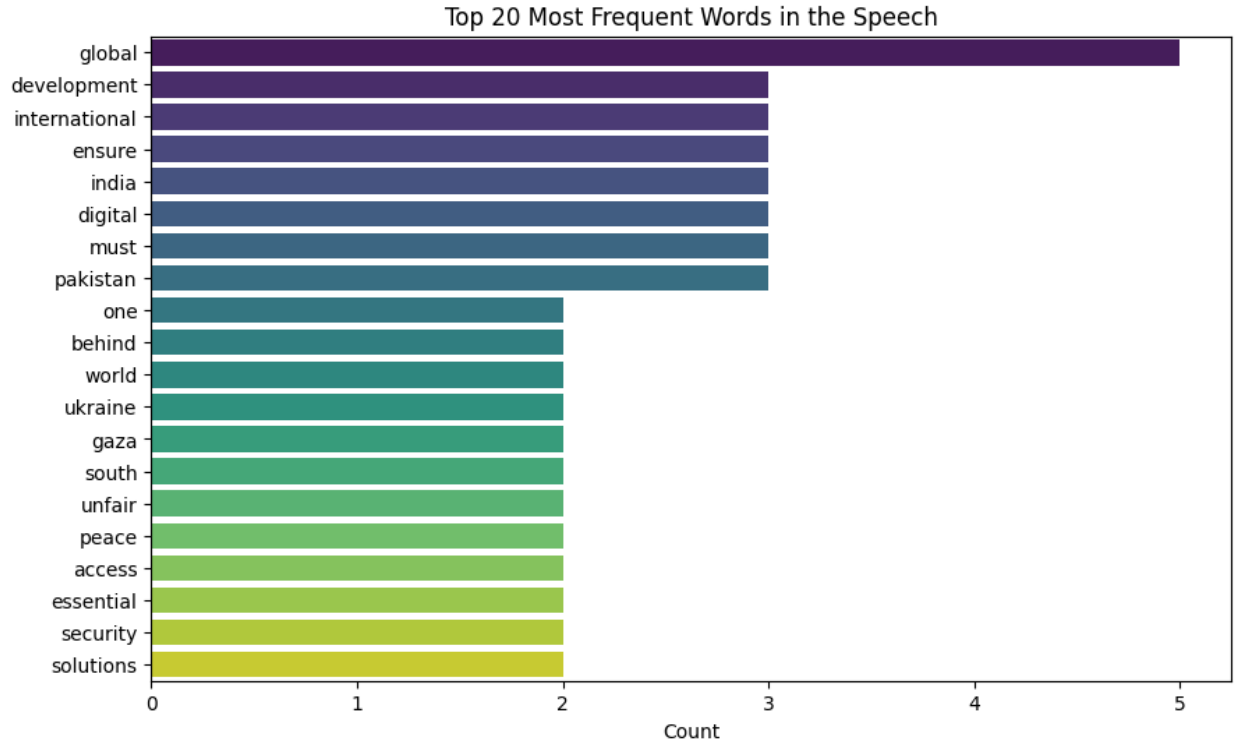
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
True
```

```
stop_words = set(stopwords.words('english'))
```

```
df['Tokenized_Details'] = df['Details'].apply(lambda x:
word_tokenize(x.lower()))
```

```
df['Cleaned_Details'] = df['Tokenized_Details'].apply(
    lambda x: [word for word in x if word.isalnum() and word not in
stop_words]
)

all_words = [word for tokens in df['Cleaned_Details'] for word in
tokens]
word_freq = Counter(all_words)

top_words = word_freq.most_common(20)
words, counts = zip(*top_words)
plt.figure(figsize=(10,6))
sns.barplot(x=list(counts), y=list(words), palette='viridis')
plt.title('Top 20 Most Frequent Words in the Speech')
plt.xlabel('Count')
plt.show()
```



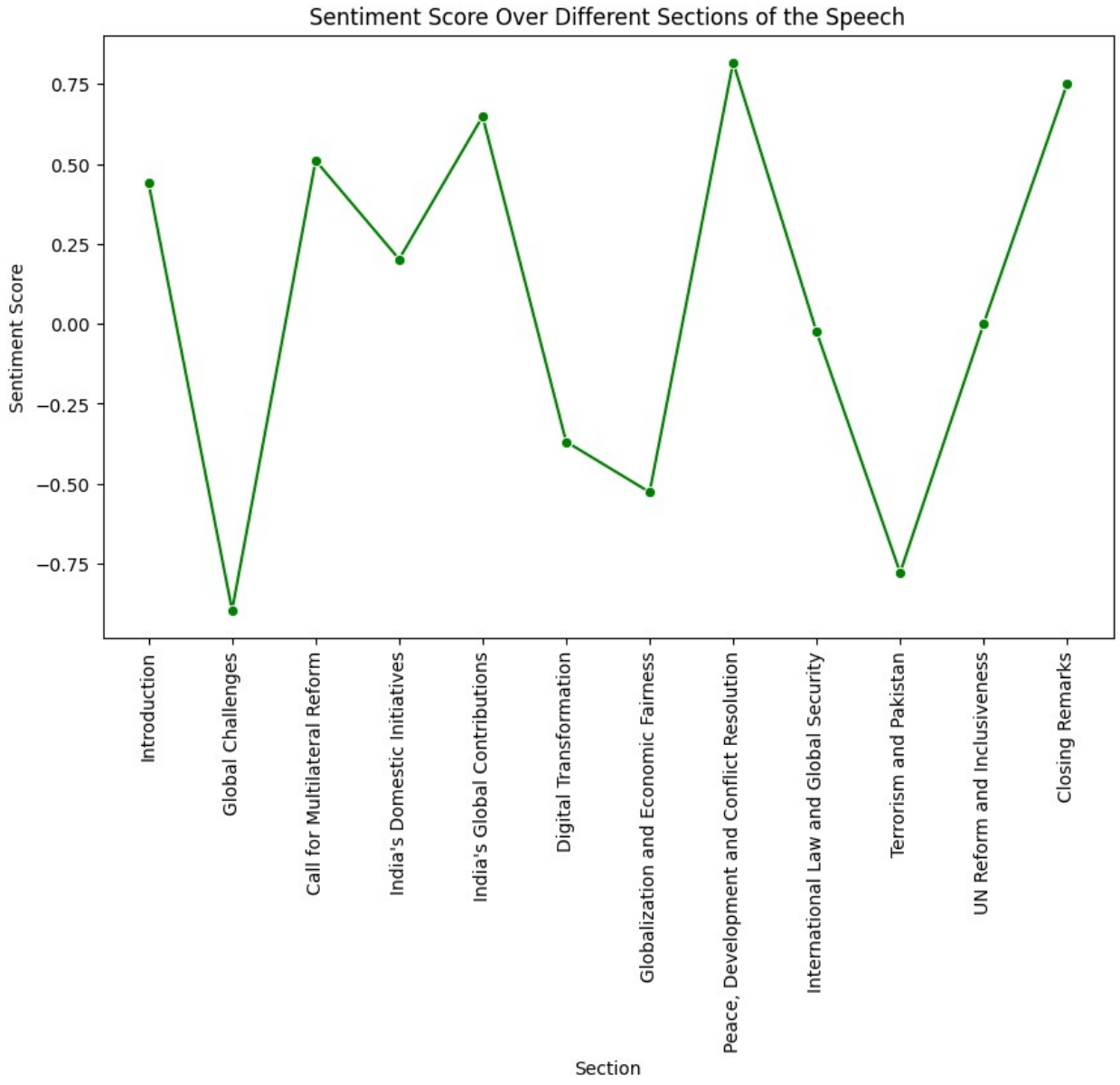

```
wordcloud = WordCloud(width=800, height=400,
background_color="white").generate_from_frequencies(word_freq)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud of the Speech")
plt.show()
```


```
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()
```

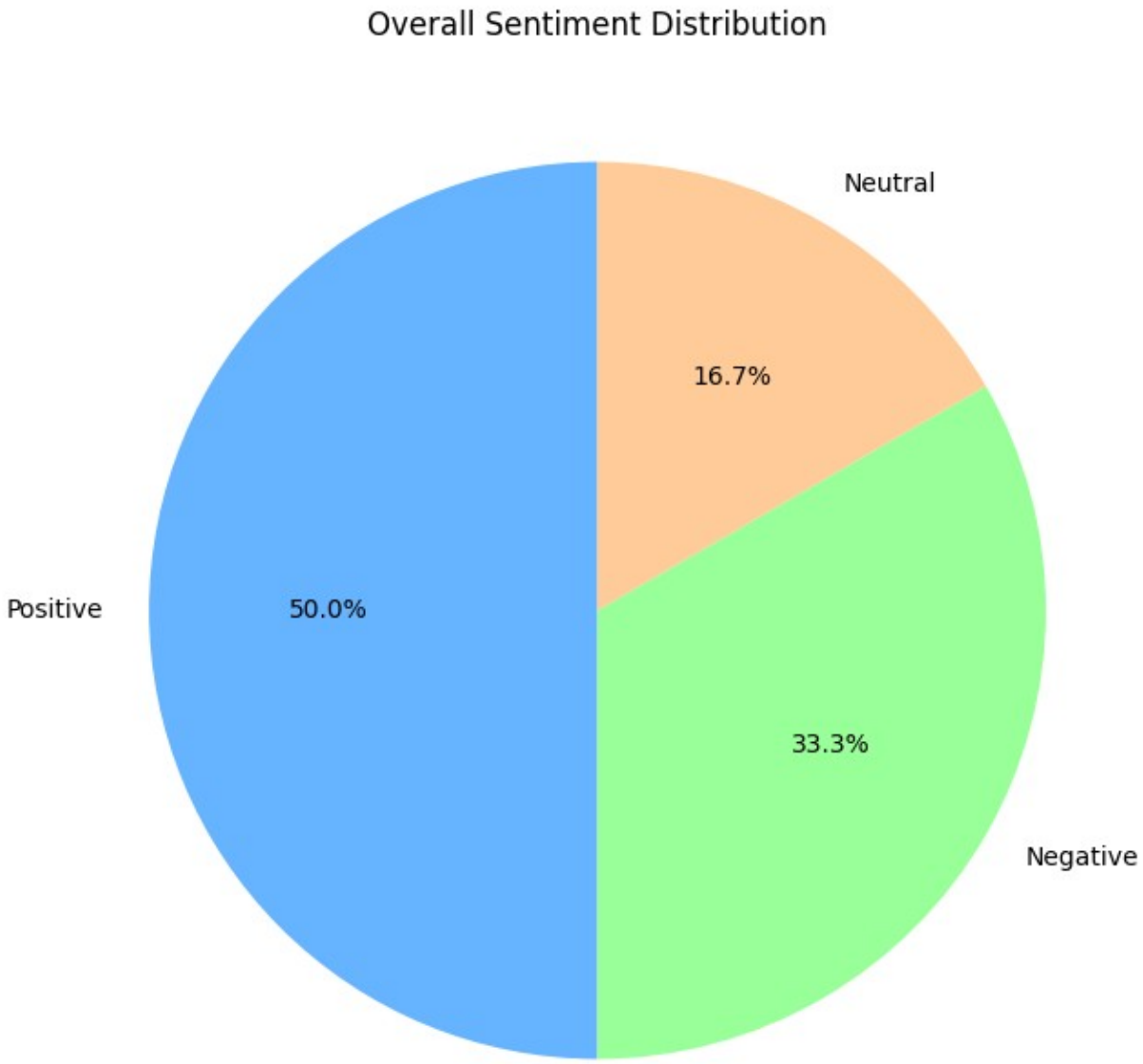
```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

```
df['Sentiment_Score'] = df['Details'].apply(lambda x:
sia.polarity_scores(x)['compound'])
```

```
plt.figure(figsize=(10,6))
sns.lineplot(x=df['Section'], y=df['Sentiment_Score'], marker='o',
color='green')
plt.xticks(rotation=90)
plt.title('Sentiment Score Over Different Sections of the Speech')
plt.ylabel('Sentiment Score')
plt.show()
```




```
df['Sentiment'] = df['Sentiment_Score'].apply(lambda x: 'Positive' if
x > 0.1 else ('Negative' if x < -0.1 else 'Neutral'))

sentiment_counts = df['Sentiment'].value_counts()

plt.figure(figsize=(8,8))
plt.pie(sentiment_counts, labels=sentiment_counts.index,
autopct='%1.1f%%', colors=['#66b3ff','#99ff99','#ffcc99'],
startangle=90)
plt.title('Overall Sentiment Distribution')
plt.show()
```

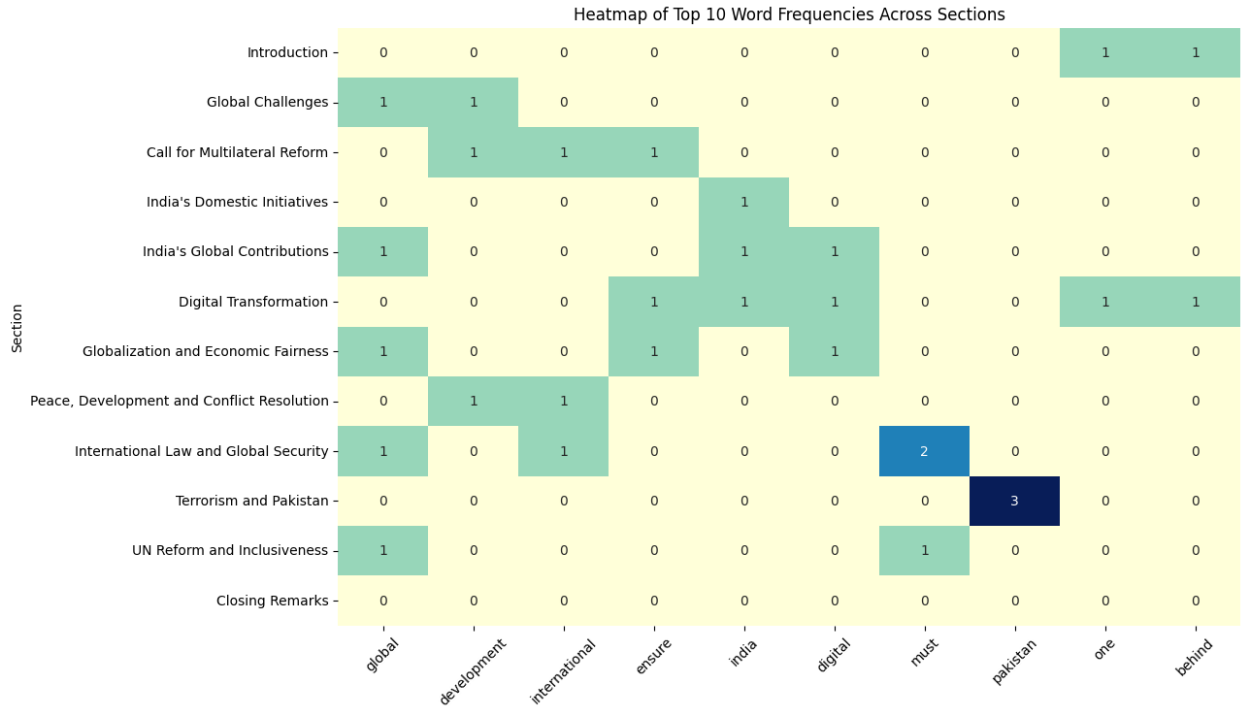



```
vectorizer = CountVectorizer(ngram_range=(2, 2), stop_words='english')
X = vectorizer.fit_transform(df['Details'])

bigrams_df = pd.DataFrame(X.toarray(),
columns=vectorizer.get_feature_names_out())
bigrams_freq = bigrams_df.sum().sort_values(ascending=False)

bigrams_freq
```

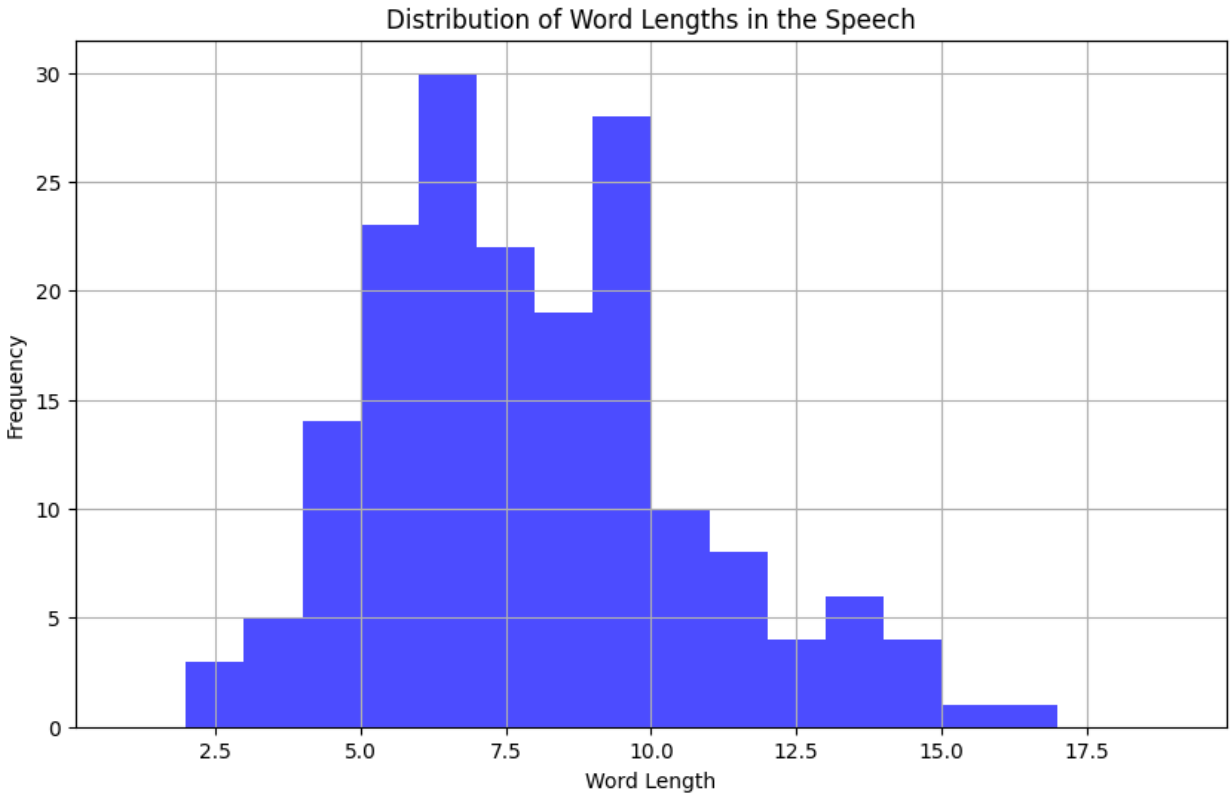
```
global south             2
radicalization clear     1
potential digital        1
```

```
practices rising         1
president supported      1
                        ..
focuses vulnerable       1
gaza development         1
gaza peace               1
global concerns          1
youth farmers            1
Length: 154, dtype: int64
```

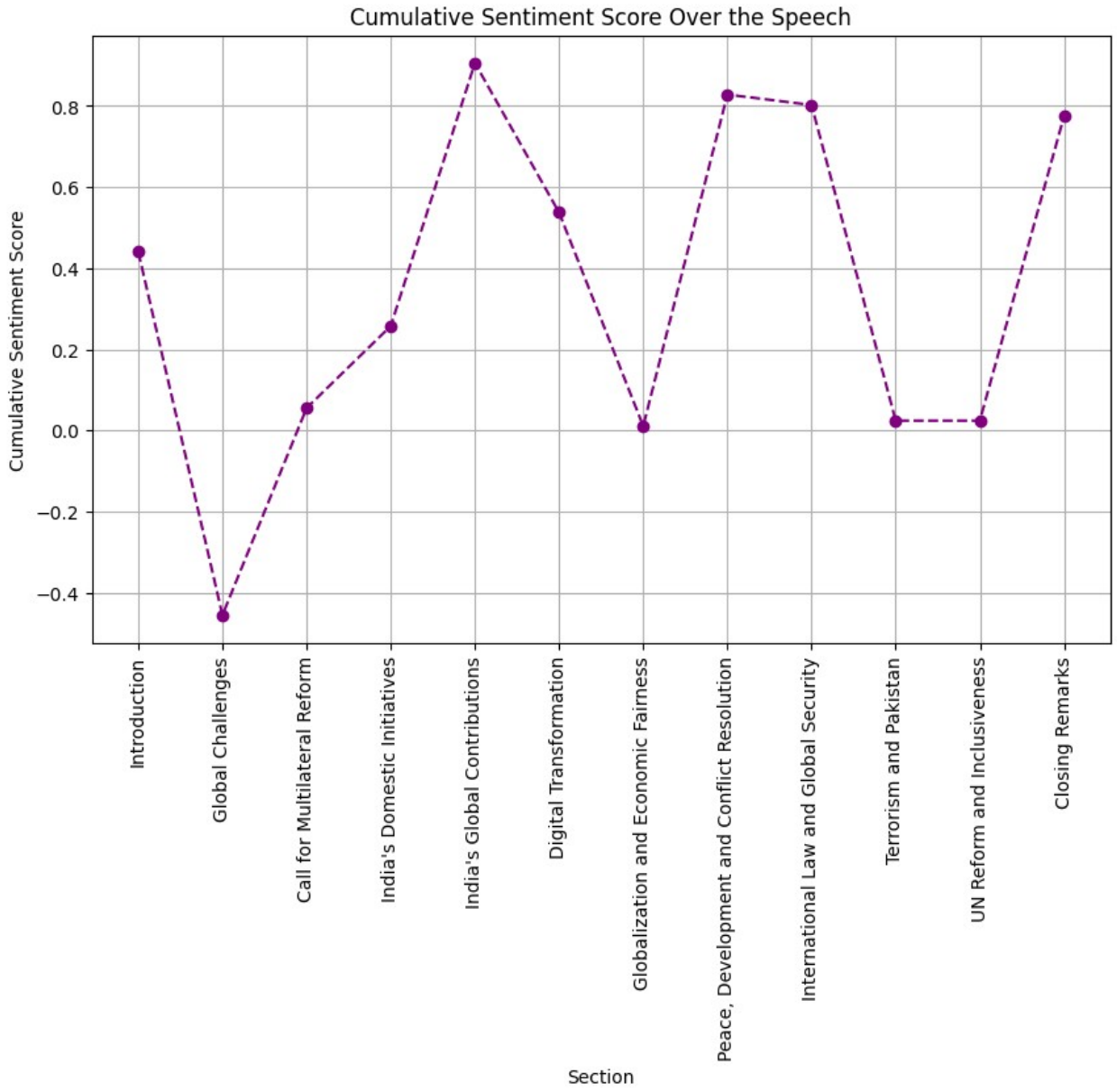
```
from nltk import pos_tag
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import TfidfVectorizer

top_10_words = [word for word, count in word_freq.most_common(10)]
word_freq_by_section = pd.DataFrame(
    {word: [tokens.count(word) for tokens in df['Cleaned_Details']]
for word in top_10_words},
    index=df['Section']
)

plt.figure(figsize=(12, 8))
sns.heatmap(word_freq_by_section, cmap='YlGnBu', annot=True, fmt='d',
cbar=False)
plt.title("Heatmap of Top 10 Word Frequencies Across Sections")
plt.xticks(rotation=45)
plt.show()
```



```
import numpy as np

word_lengths = [len(word) for word in all_words]
plt.figure(figsize=(10,6))
plt.hist(word_lengths, bins=np.arange(1, 20), color='blue', alpha=0.7)
plt.title("Distribution of Word Lengths in the Speech")
plt.xlabel("Word Length")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()
```

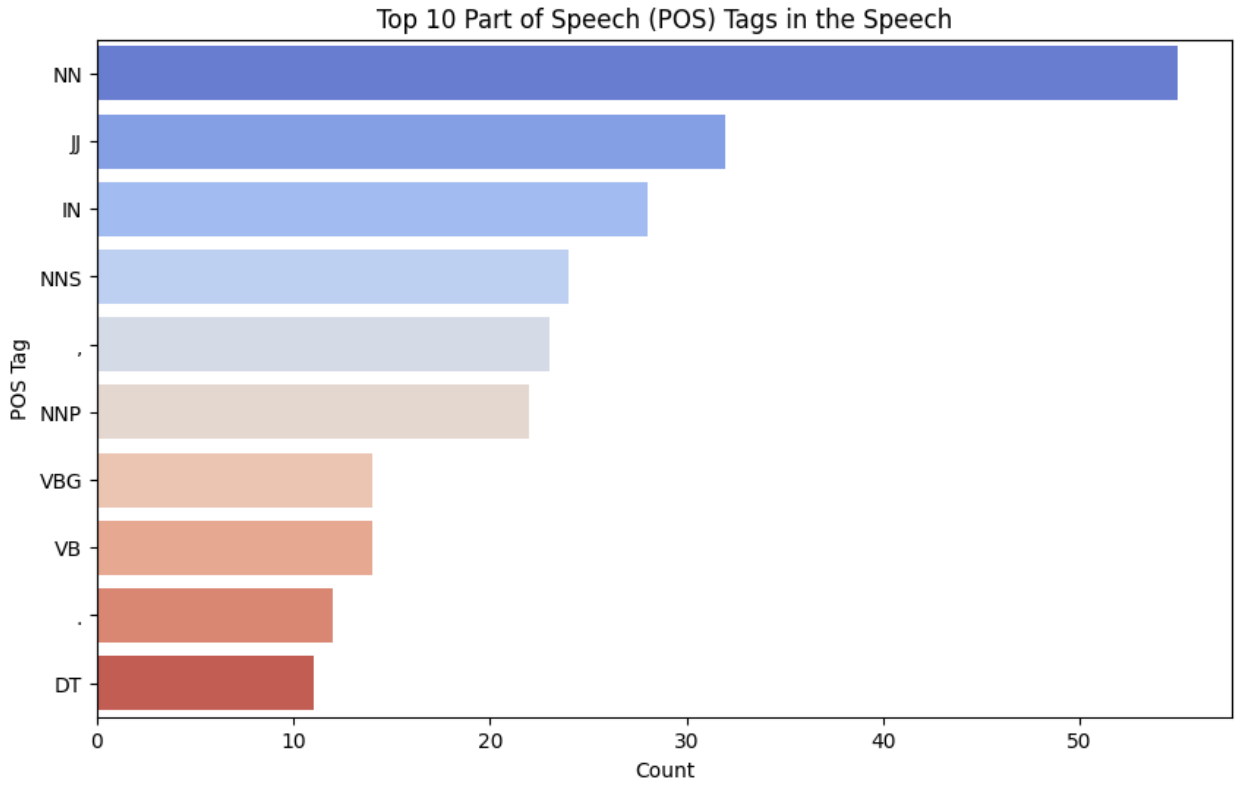



```
df['Cumulative_Sentiment'] = df['Sentiment_Score'].cumsum()

plt.figure(figsize=(10,6))
plt.plot(df['Section'], df['Cumulative_Sentiment'], marker='o',
linestyle='--', color='purple')
plt.xticks(rotation=90)
plt.title("Cumulative Sentiment Score Over the Speech")
plt.xlabel("Section")
plt.ylabel("Cumulative Sentiment Score")
plt.grid(True)
plt.show()
```

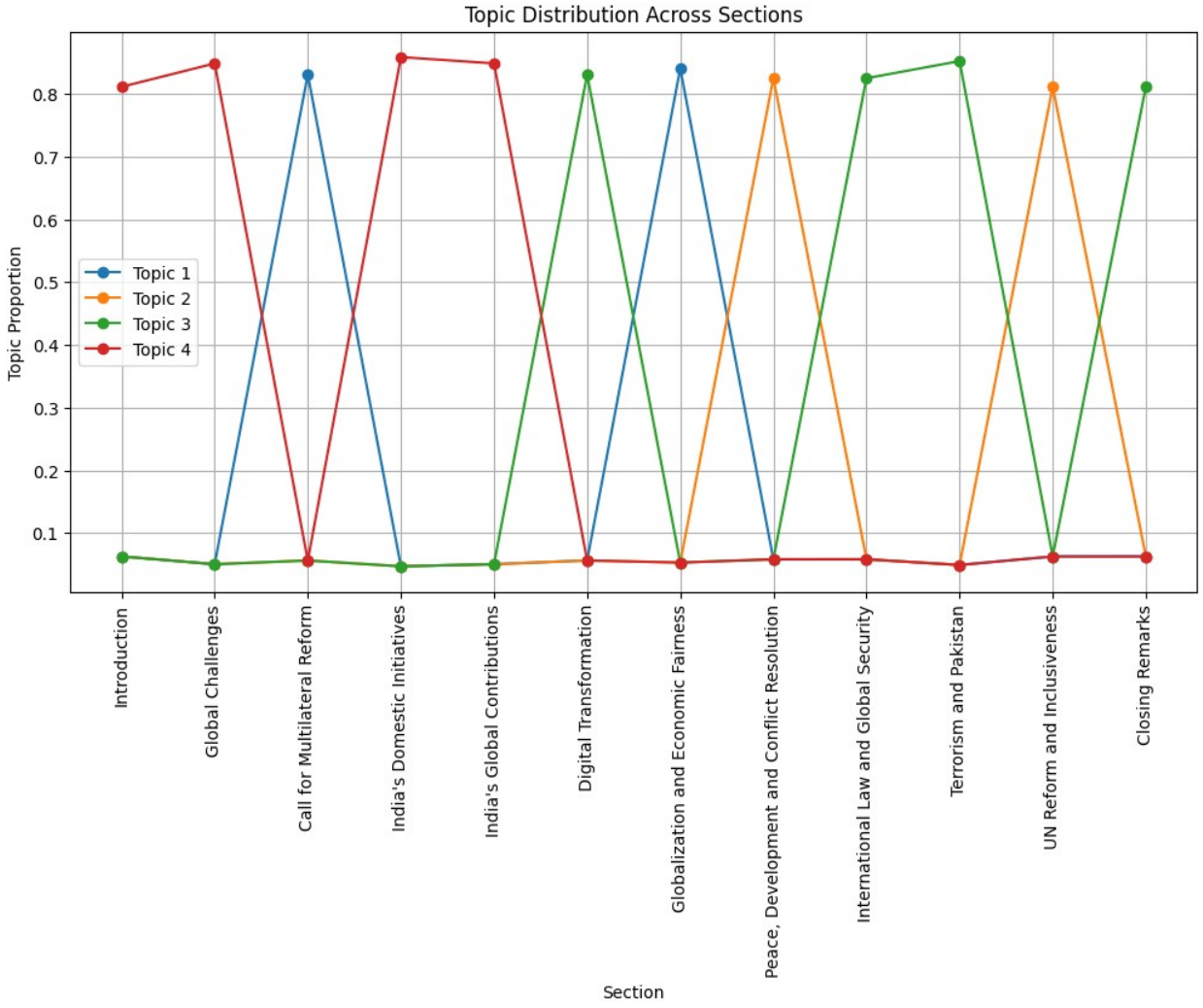



```
df['POS_Tags'] = df['Details'].apply(lambda x:
pos_tag(word_tokenize(x)))
all_pos_tags = [tag for pos_list in df['POS_Tags'] for _, tag in
pos_list]

pos_counts = Counter(all_pos_tags)
pos_counts = pos_counts.most_common(10)

plt.figure(figsize=(10,6))
sns.barplot(x=[count for _, count in pos_counts], y=[tag for tag, _ in
pos_counts], palette="coolwarm")
plt.title("Top 10 Part of Speech (POS) Tags in the Speech")
plt.xlabel("Count")
```

```
plt.ylabel("POS Tag")
plt.show()
```




```
tfidf_vectorizer = TfidfVectorizer(ngram_range=(2,2),
stop_words='english')
X_tfidf = tfidf_vectorizer.fit_transform(df['Details'])

lda_model = LatentDirichletAllocation(n_components=4, random_state=42)
lda_topics = lda_model.fit_transform(X_tfidf)

plt.figure(figsize=(12, 6))
for topic_idx in range(lda_topics.shape[1]):
    plt.plot(df['Section'], lda_topics[:, topic_idx], marker='o',
label=f"Topic {topic_idx + 1}")

plt.xticks(rotation=90)
plt.title("Topic Distribution Across Sections")
plt.xlabel("Section")
plt.ylabel("Topic Proportion")
plt.legend()
plt.grid(True)
plt.show()
```




```
import spacy
from collections import Counter

nlp = spacy.load("en_core_web_sm")

doc = nlp(' '.join(df['Details']))

entities = [(ent.text, ent.label_) for ent in doc.ents]

entity_df = pd.DataFrame(entities, columns=['Entity', 'Label'])

entity_df
```

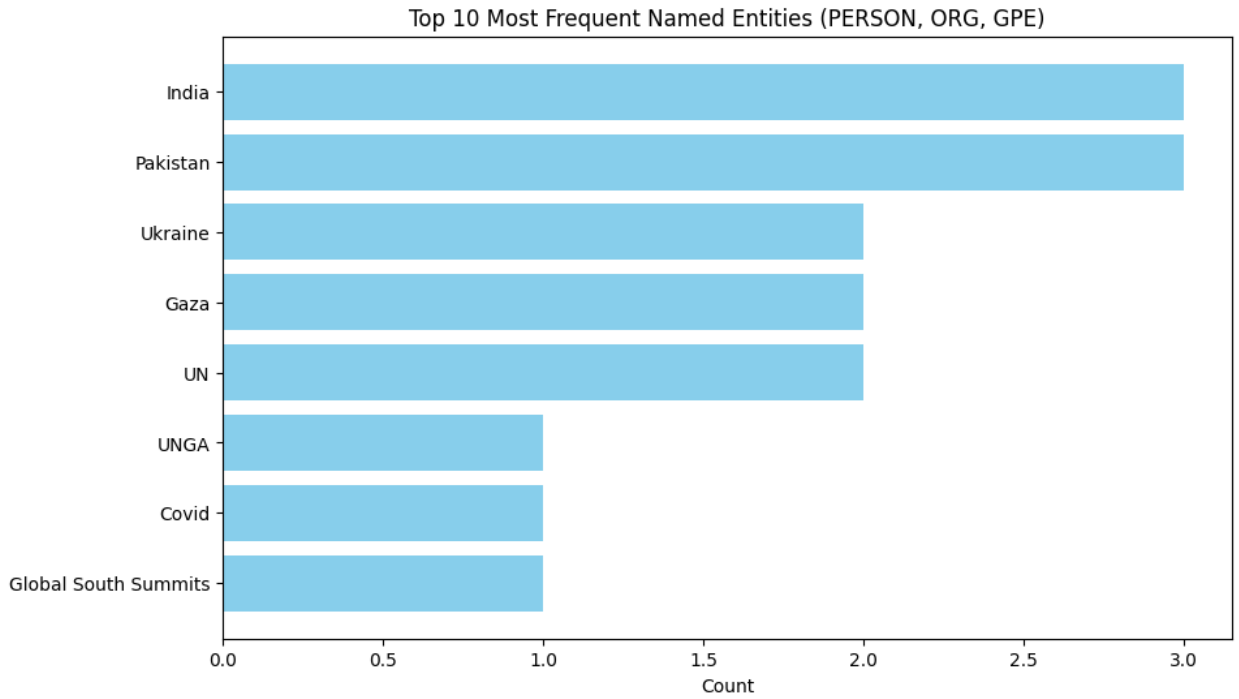
	Entity	Label
0	1.4 billion	CARDINAL
1	Bharat	NORP
2	UNGA	ORG
3	Covid	GPE
4	Ukraine	GPE

```
5                    Gaza       GPE
6            Global South       LOC
7                   India       GPE
8                   India       GPE
9                      78  CARDINAL
10  Global South Summits       ORG
11                  India       GPE
12                Ukraine       GPE
13                   Gaza       GPE
14               Pakistan       GPE
15               Pakistan       GPE
16               Pakistan       GPE
17                 Indian      NORP
18                     UN       ORG
19                     UN       ORG
```

```
entity_count = entity_df['Label'].value_counts()

entity_count
```

```
Label
GPE         11
ORG          4
CARDINAL     2
NORP         2
LOC          1
Name: count, dtype: int64
```

```
filtered_entities = entity_df[entity_df['Label'].isin(['PERSON',
'ORG', 'GPE'])]

top_entities = Counter(filtered_entities['Entity']).most_common(10)

top_entities_df = pd.DataFrame(top_entities, columns=['Entity',
'Count'])

plt.figure(figsize=(10,6))
plt.barh(top_entities_df['Entity'], top_entities_df['Count'],
color='skyblue')
plt.xlabel('Count')
plt.title('Top 10 Most Frequent Named Entities (PERSON, ORG, GPE)')
plt.gca().invert_yaxis()
plt.show()
```




```
from spacy import displacy

for sent in doc.sents:
    displacy.render(nlp(sent.text), style='ent', jupyter=True)
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>
```

Thanks !!!