bharat-g20

September 10, 2023

Importing Libraries

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
```

[3]:
```
import warnings
warnings.filterwarnings('ignore')
```

Loading data

[17]:
```
file_path = "contents\Prime Minister's remarks.txt"
```

[18]:
```
with open(file_path , 'r' , encoding = "utf-8") as file:
  speech_text = file.read()
```

NLP comes into the Picture

[19]:
```
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

Downloading the necessary batches

[20]:
```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('vader_lexicon')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to C:\Users\TANMAY SINGH
[nltk_data]     THAKUR\AppData\Roaming\nltk_data…
```

```
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to C:\Users\TANMAY SINGH
[nltk_data]     THAKUR\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package vader_lexicon to C:\Users\TANMAY SINGH
[nltk_data]     THAKUR\AppData\Roaming\nltk_data…
[nltk_data] Downloading package wordnet to C:\Users\TANMAY SINGH
[nltk_data]     THAKUR\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[20]:
```
True
```

Performing NLP operations

[21]:
```
# Remove non-alphanumeric characters and whitespace from the speech text.
speech_text_cleaned = re.sub(r'[^\w\s]', '', speech_text)

# Convert the cleaned text to lowercase for consistency.
speech_text_cleaned = speech_text_cleaned.lower()

# Tokenize the cleaned text into words.
words = word_tokenize(speech_text_cleaned)

# Load the English stop words (common words that may not carry significant␣
 ↪meaning).
stop_words = set(stopwords.words('english'))

# Filter out the stop words from the list of words to focus on meaningful␣
 ↪content.
words_filtered = [word for word in words if word not in stop_words]
```

[22]:
```
# Initialize a WordNet lemmatizer, which reduces words to their base or␣
 ↪dictionary form.
lemmatizer = WordNetLemmatizer()

# Lemmatize the filtered words to convert them to their base forms.
words_lemmatized = [lemmatizer.lemmatize(word) for word in words_filtered]
```

Getting average sentiment

[23]:
```
# Initialize the SentimentIntensityAnalyzer from the NLTK library.
sia = SentimentIntensityAnalyzer()

# Calculate sentiment scores for each lemmatized word and store them in a list.
sentiment_scores = [sia.polarity_scores(word)["compound"] for word in␣
 ↪words_lemmatized]
```

```
# Calculate the average sentiment score by summing up all scores and dividing␣
 ↪by the number of scores.
average_sentiment = sum(sentiment_scores) / len(sentiment_scores)
```

[24]:
```
print("The average sentiment is :" , average_sentiment)
```

```
The average sentiment is : 0.04182883613202077
```

EXTRACT POSITIVE NEGATIVE AND NEUTRAL

[25]:
```
# Create a list of words with positive sentiment scores (score > 0.1).
positive_words = [word for i, word in enumerate(words_filtered) if␣
 ↪sentiment_scores[i] > 0.1]

# Create a list of words with negative sentiment scores (score < -0.1).
negative_words = [word for i, word in enumerate(words_filtered) if␣
 ↪sentiment_scores[i] < -0.1]

# Create a list of words with neutral sentiment scores (score between -0.1 and␣
 ↪0.1).
neutral_words = [word for i, word in enumerate(words_filtered) if -0.1 <=␣
 ↪sentiment_scores[i] <= 0.1]
```

[26]:
```
# Print the list of words with positive sentiment scores.
print('The positive words are:', positive_words)
```

```
The positive words are: ['excellencies', 'like', 'heartfelt', 'pray', 'well',
'ready', 'excellencies', 'extends', 'warm', 'welcome', 'happiness', 'solutions',
'friends', 'huge', 'trust', 'trust', 'trust', 'invites', 'trust', 'trust',
'confidence', 'ensuring', 'energy', 'security', 'solutions', 'friends',
'spirit', 'engaged', 'spirit', 'agree', 'consent', 'invite', 'friends', 'faith',
'respect', 'spirit', 'support', 'celebrating', 'security', 'spirit', 'launched',
'natural', 'protect', 'well', 'launched', 'boost', 'significant', 'creating',
'friends', 'energy', 'significant', 'energy', 'play', 'pleased', 'positive',
'willingness', 'fulfill', 'commitment', 'commitments', 'growth', 'friends',
'spirit', 'greater', 'good', 'stable', 'energy', 'security', 'invites', 'join',
'friends', 'credit', 'credit', 'positive', 'adequate', 'encouragement',
'positive', 'credit', 'promote', 'positive', 'credit', 'friends', 'success',
'beneficial', 'spirit', 'shared', 'invites', 'join', 'friends', 'warm',
'welcome', 'congratulations', 'eager', 'friends', 'good', 'support', 'agreed',
'adopt', 'adopt', 'like', 'congratulate', 'worthwhile', 'excellencies',
'creation', 'innovation', 'solutions', 'like', 'support', 'support',
'happiness', 'happy', 'spirit', 'share', 'huge', 'certainly', 'beneficial',
'friends', 'talent', 'good', 'talent', 'growth', 'friends', 'huge', 'promotes',
'trust', 'special', 'help', 'strengthen', 'active', 'friends', 'pleased',
'agreement', 'increase', 'gratitude', 'friends', 'wellness', 'strengthen',
'wellness', 'hope', 'friends', 'help', 'significant', 'friends', 'like', 'top',
'talent', 'special', 'beneficial', 'top', 'talent', 'opportunities', 'talent',
```

```
'like', 'great', 'satisfaction', 'friends', 'like', 'excellencies', 'satisfied',
'optimistic', 'vision', 'interests', 'friends', 'vision', 'like', 'sharing',
'sharing', 'interest', 'commitment', 'growth', 'happy', 'strong', 'agreed',
'accepted', 'engagement', 'friends', 'intelligence', 'responsible', 'benefits',
'like', 'friends', 'security', 'security', 'care', 'security', 'feeling',
'strengthened', 'friends', 'better', 'united', 'security', 'respect',
'effective', 'friends', 'increase', 'expand', 'effective', 'friends',
'excellencies', 'like', 'friends', 'faith', 'spirit', 'support', 'congratulate',
'friend', 'invite', 'share', 'excellencies', 'progress', 'share', 'hope',
'join', 'excellencies', 'pleasant', 'hope', 'peace', 'sincere', 'gratitude']
```

[27]:
```
# Print the list of words with negative sentiment scores.
print('The negative words are:', negative_words)
```

```
The negative words are: ['affected', 'injured', 'difficult', 'demand', 'crisis',
'lack', 'conflict', 'deficit', 'crisis', 'deficit', 'terrorism', 'negative',
'lack', 'hard', 'sorrow', 'sad', 'pay', 'debt', 'crisis', 'crisis', 'crises',
'critical', 'disorders', 'cancer', 'problems', 'terrorism', 'exploiting',
'critical', 'fail', 'lose']
```

[28]:
```
# Print the list of words with neutral sentiment scores.
print('The neutral words are:', neutral_words)
```

```
The neutral words are: ['prime', 'ministers', 'opening', 'remarks', 'g20',
'summit', 'september', '08', '2023', 'highnesses', 'namaskar', 'starting',
'formal', 'proceedings', 'behalf', 'us', 'would', 'express', 'condolences',
'people', 'earthquake', 'occurred', 'morocco', 'moment', 'ago', 'people', 'get',
'soon', 'entire', 'world', 'community', 'morocco', 'time', 'provide',
'possible', 'assistance', 'highnesses', 'president', 'g20', 'india', 'place',
'gathered', 'today', 'kilometers', 'away', 'stands', 'pillar', 'nearly', 'two',
'half', 'thousand', 'years', 'old', 'inscribed', 'pillar', 'prakrit',
'language', 'words', '  ', '  ', '  ', ' ', ' ', '  ', '  ', '  ',
'meaning', 'welfare', 'humanity', 'always', 'ensured', 'two', 'half',
'thousand', 'years', 'ago', 'land', 'india', 'gave', 'message', 'entire',
'world', 'let', 'us', 'begin', 'g20', 'summit', 'remembering', 'message',
'21st', 'century', 'time', 'potential', 'give', 'new', 'direction', 'entire',
'world', 'time', 'years', 'old', 'challenges', 'new', 'us', 'therefore', 'must',
'move', 'forward', 'fulfilling', 'responsibilities', 'human', 'centric',
'approach', 'covid19', 'come', 'world', 'deepened', 'overcome', 'covid', 'also',
'overcome', 'mutual', 'today', 'president', 'g20', 'india', 'entire', 'world',
'come', 'together', 'first', 'foremost', 'transform', 'global', 'global',
'time', 'us', 'walk', 'together', 'mantra', 'sabka', 'saath', 'sabka', 'vikas',
'sabka', 'vishwas', 'sabka', 'prayas', 'become', 'guiding', 'light', 'us',
'whether', 'turbulent', 'global', 'economy', 'northsouth', 'divide', 'distance',
'east', 'west', 'management', 'food', 'fuel', 'fertilizer', 'dealing',
'cybersecurity', 'health', 'water', 'must', 'move', 'towards', 'concrete',
'challenges', 'present', 'also', 'future', 'generations', 'indias', 'g20',
'presidency', 'become', 'symbol', 'inclusion', 'within', 'country', 'beyond',
```

'representing', 'sabka', 'saath', 'become', 'peoples', 'g20', 'millions',
'indians', '200', 'meetings', 'held', '60', 'cities', 'across', 'country',
'sabka', 'saath', 'india', 'proposed', 'permanent', 'membership', 'african',
'union', 'g20', 'believe', 'proposal', 'proceed', 'proceedings', 'chairperson',
'african', 'union', 'take', 'place', 'permanent', 'member', 'g20', 'prime',
'ministers', 'remarks', 'g20', 'summit', 'session', '1', 'september', '09',
'2023', 'india', 'land', 'diversity', 'spirituality', 'traditions', 'many',
'major', 'religions', 'world', 'born', 'every', 'religion', 'world', 'found',
'mother', 'democracy', 'belief', 'dialogue', 'democratic', 'principles',
'unwavering', 'since', 'time', 'immemorial', 'global', 'conduct', 'rooted',
'fundamental', 'principle', 'vasudhaiva', 'kutumbakam', 'means', 'world', 'one',
'family', 'notion', 'considering', 'world', 'one', 'family', 'also', 'connects',
'every', 'indian', 'sense', 'responsibility', 'one', 'earth', 'one', 'earth',
'india', 'initiated', 'lifestyle', 'environment', 'mission', 'indias',
'initiative', 'entire', 'world', 'international', 'year', 'millets', 'year',
'alignment', 'principles', 'climate', 'line', 'india', 'green', 'grids',
'initiative', 'one', 'sun', 'one', 'world', 'one', 'grid', 'cop26', 'today',
'india', 'stands', 'among', 'countries', 'largescale', 'solar', 'revolution',
'underway', 'millions', 'indian', 'farmers', 'embraced', 'farming', 'big',
'campaign', 'human', 'health', 'health', 'soil', 'earth', 'also', 'national',
'green', 'hydrogen', 'mission', 'india', 'green', 'hydrogen', 'production',
'indias', 'g20', 'presidency', 'also', 'taken', 'steps', 'toward', 'global',
'hydrogen', 'ecosystem', 'keeping', 'mind', 'challenge', 'climate', 'change',
'transition', 'need', '21st', 'century', 'world', 'trillions', 'dollars',
'required', 'inclusive', 'transition', 'naturally', 'developed', 'countries',
'crucial', 'role', 'along', 'india', 'countries', 'global', 'south',
'developed', 'countries', 'taken', 'initiative', 'year', '2023', 'developed',
'countries', 'expressed', '100', 'billion', 'dollars', 'climate', 'finance',
'first', 'time', 'adopting', 'green', 'development', 'pact', 'g20', 'also',
'reaffirmed', 'sustainable', 'green', 'collective', 'effort', 'today', 'india',
'suggestions', 'g20', 'platform', 'today', 'need', 'hour', 'countries', 'work',
'together', 'field', 'fuel', 'blending', 'proposal', 'take', 'initiative',
'globallevel', 'take', 'ethanol', 'blending', 'petrol', '20', 'percent',
'alternatively', 'could', 'work', 'developing', 'another', 'blending', 'mix',
'global', 'one', 'ensures', 'supply', 'also', 'contributing', 'climate',
'context', 'today', 'launching', 'global', 'biofuel', 'alliance', 'india',
'initiative', 'keeping', 'environment', 'mind', 'discussions', 'carbon',
'ongoing', 'decades', 'carbon', 'emphasizes', 'done', 'perspective', 'result',
'steps', 'taken', 'often', 'receive', 'attention', 'initiatives', 'green',
'shows', 'us', 'way', 'forward', 'thinking', 'propose', 'g20', 'countries',
'start', 'working', 'green', 'initiative', 'familiar', 'indias', 'moon',
'mission', 'chandrayaan', 'data', 'obtained', 'humanity', 'india', 'proposing',
'launch', 'g20', 'satellite', 'mission', 'environment', 'climate',
'observation', 'climate', 'weather', 'data', 'obtained', 'countries',
'especially', 'countries', 'global', 'south', 'india', 'g20', 'countries',
'initiative', 'hear', 'thoughts', 'prime', 'ministers', 'remarks', 'g20',
'summit', 'session', '2', 'september', '09', '2023', 'received', 'news', 'due',
'work', 'teams', 'new', 'delhi', 'g20', 'leaders', 'summit', 'declaration',

'propose', 'also', 'leaders', 'declaration', 'declare', 'declaration',
'occasion', 'would', 'minister', 'sherpa', 'officials', 'put', 'immense',
'effort', 'make', 'therefore', 'deserve', 'congratulated', 'highness',
'ancient', 'vedas', 'composed', 'thousands', 'years', 'ago', 'said', ' ',
' ', ' ', 'means', 'one', 'let', 'become', 'many', 'need', 'move',
'viable', 'means', 'thinking', 'whole', 'self', 'wellbeing', 'us', 'instead',
'emphasize', 'must', 'connect', 'every', 'class', 'every', 'country', 'every',
'society', 'every', 'region', 'world', 'essence', 'one', 'family', 'concept',
'every', 'family', 'system', 'need', 'build', 'global', 'system', 'together',
'develop', 'mindset', 'someone', 'elses', 'make', 'us', 'someone', 'elses',
'make', 'us', 'equally', 'think', 'one', 'family', 'also', 'keep', 'mind',
'empower', 'member', 'india', 'wants', 'every', 'experience', 'global',
'family', 'india', 'adopted', 'technology', 'bridge', 'make', 'development',
'inclusive', 'sustainable', 'india', 'developed', 'new', 'model', 'inclusion',
'transparency', 'targeted', 'interventions', 'jam', 'trinity', 'bank',
'accounts', 'aadhar', 'identity', 'mobile', 'phones', 'world', 'bank', 'also',
'acknowledged', 'jam', 'trinity', 'achieved', 'financial', 'inclusion', 'rate',
'6', 'years', 'would', 'taken', '47', 'years', 'achieve', 'using', 'model',
'india', 'transferred', '360', 'billion', 'dollars', 'directly', 'bank',
'accounts', 'need', 'past', 'decade', 'prevented', 'leakage', 'approximately',
'33', 'billion', 'dollars', 'nearly', '125', 'gdp', 'model', 'global', 'family',
'especially', 'global', 'south', 'indias', 'youth', 'young', 'also', 'global',
'form', 'one', 'family', 'times', 'come', 'large', 'skilled', 'young', 'pool',
'crucial', 'sustain', 'global', 'therefore', 'move', 'towards', 'global',
'skill', 'mapping', 'also', 'priority', 'global', 'south', 'talking', 'one',
'family', 'must', 'also', 'keep', 'mind', 'challenges', 'lie', 'ahead',
'global', 'family', 'seen', 'global', 'challenge', 'came', 'form', 'covid',
'global', 'supply', 'chains', 'built', 'decades', 'completely', 'exposed',
'concept', 'one', 'family', 'today', 'must', 'build', 'global', 'supply',
'chain', 'transparency', 'collective', 'responsibility', 'view', 'countries',
'humanity', 'solely', 'markets', 'need', 'sensitive', 'longterm', 'approach',
'must', 'attention', 'capacity', 'building', 'developing', 'countries',
'therefore', 'indias', 'proposal', 'mapping', 'framework', 'existing', 'supply',
'chain', 'make', 'global', 'supply', 'chain', 'inclusive', 'must', 'also',
'acknowledge', 'role', 'small', 'businesses', 'essential', 'access', 'markets',
'information', 'trade', 'costs', 'reduced', 'continuing', 'one', 'family',
'mantra', 'must', 'also', 'address', 'faced', 'developing', 'countries',
'sensitivity', 'need', 'establish', 'system', 'allows', 'countries', 'overcome',
'ensures', 'never', 'happen', 'im', 'action', 'plan', 'accelerate',
'sustainable', 'development', 'goals', 'financing', 'purpose', 'express', 'one',
'family', 'approach', 'equally', 'essential', 'holistic', 'health', 'system',
'global', 'centre', 'traditional', 'medicine', 'established', 'india',
'promotion', 'worldwide', 'soon', 'make', 'efforts', 'establish', 'global',
'repository', 'traditional', 'medicine', 'every', 'society', 'around', 'world',
'mothers', 'driving', 'force', 'family', 'today', 'india', 'see', 'womens',
'leadership', 'every', 'sector', 'approximately', '45', 'stem', 'science',
'technology', 'engineering', 'mathematics', 'graduates', 'india', 'women',
'today', 'many', 'missions', 'indias', 'space', 'program', 'handled', 'female',

'scientists', 'today', 'around', '90', 'million', 'women', 'indian', 'villages',
'taking', 'forward', 'small', 'businesses', 'joining', 'self', 'groups',
'campaign', 'believe', 'womenled', 'development', 'driver', 'change', '21st',
'century', 'one', 'family', 'session', 'would', 'present', 'three',
'suggestions', 'first', 'urge', 'sports', 'leagues', 'world', 'invest', '5',
'earnings', 'sports', 'infrastructure', 'women', 'countries', 'global', 'south',
'could', 'serve', 'new', 'model', 'publicprivate', 'partnership', 'global',
'level', 'second', 'similar', 'countries', 'issue', 'different', 'categories',
'visas', 'establish', 'g20', 'visa', 'category', 'type', 'visa', 'could',
'highly', 'science', 'technology', 'explore', 'global', 'efforts', 'contribute',
'significantly', 'economies', 'third', 'consider', 'establishment', 'global',
'biobanks', 'supervision', 'biobanks', 'could', 'focus', 'specifically',
'diseases', 'heart', 'diseases', 'sickle', 'cell', 'anemia', 'endocrine',
'breast', 'setting', 'global', 'biobanks', 'india', 'would', 'bring', 'us',
'would', 'hear', 'prime', 'ministers', 'remarks', 'g20', 'summit', 'session',
'3', 'highnesses', 'yesterday', 'extensive', 'discussions', 'one', 'earth',
'one', 'family', 'sessions', 'today', 'g20', 'become', 'platform', 'efforts',
'regarding', 'one', 'earth', 'one', 'family', 'one', 'future', 'currently',
'discussing', 'future', 'transcend', 'concept', 'global', 'village', 'witness',
'global', 'family', 'becoming', 'reality', 'future', 'countries', 'intertwined',
'also', 'hearts', 'interconnected', 'continuously', 'drawn', 'attention',
'human', 'centric', 'instead', 'gdp', 'centric', 'approach', 'today', 'many',
'countries', 'india', 'much', 'whole', 'world', 'india', 'talked', 'data',
'chandrayaan', 'mission', 'everyone', 'humanity', 'also', 'proof', 'towards',
'human', 'centric', 'india', 'harnessed', 'technology', 'foster', 'inclusive',
'development', 'facilitate', 'lastmile', 'delivery', 'even', 'remote',
'villages', 'smallscale', 'traders', 'using', 'digital', 'payments', 'indias',
'chairmanship', 'framework', 'digital', 'public', 'infrastructure', 'upon',
'similarly', 'g20', 'principles', 'harnessing', 'data', 'development', 'also',
'also', 'decided', 'launch', 'data', 'development', 'capacity', 'building',
'initiative', 'development', 'global', 'south', 'formation', 'startup', '20',
'group', 'presidency', 'india', 'also', 'big', 'step', 'today', 'witnessing',
'unprecedented', 'scale', 'speed', 'newgeneration', 'technologies',
'artificial', 'example', 'front', 'us', '2019', 'adopted', 'principles', 'ai',
'today', 'need', 'take', 'one', 'step', 'propose', 'establish', 'framework',
'humancentric', 'ai', 'governance', 'india', 'also', 'give', 'suggestions',
'regard', 'endeavor', 'countries', 'get', 'ai', 'areas', 'socioeconomic',
'development', 'global', 'workforce', 'rd', 'today', 'world', 'also', 'facing',
'burning', 'affecting', 'present', 'future', 'countries', 'familiar',
'challenges', 'cyber', 'cryptocurrency', 'field', 'cryptocurrency', 'emerged',
'new', 'topic', 'everyone', 'ie', 'social', 'order', 'monetary', 'financial',
'stability', 'therefore', 'develop', 'global', 'standards', 'regulate',
'cryptocurrencies', 'basel', 'standards', 'bank', 'regulation', 'front', 'us',
'model', 'imperative', 'take', 'concrete', 'steps', 'direction', 'soon',
'possible', 'likewise', 'global', 'cooperation', 'framework', 'cybersecurity',
'essential', 'new', 'channels', 'new', 'funding', 'methods', 'cyber', 'world',
'making', 'issue', 'prosperity', 'every', 'nation', 'take', 'every', 'country',
'sensitivity', 'every', 'country', 'one', 'future', 'take', 'world', 'towards',

```
'future', 'necessary', 'global', 'systems', 'accordance', 'realities',
'present', 'today', 'nations', 'council', 'also', 'example', 'un',
'established', 'world', 'time', 'completely', 'different', 'today', 'time',
'51', 'founding', 'members', 'un', 'today', 'number', 'countries', 'included',
'un', 'around', '200', 'despite', 'permanent', 'members', 'unsc', 'still',
'till', 'today', 'world', 'changed', 'lot', 'every', 'transport',
'communication', 'health', 'education', 'every', 'sector', 'transformed', 'new',
'realities', 'reflected', 'new', 'global', 'structure', 'law', 'nature',
'individuals', 'organizations', 'adapt', 'changing', 'times', 'inevitably',
'relevance', 'must', 'think', 'open', 'mind', 'reason', 'many', 'regional',
'forums', 'come', 'existence', 'past', 'years', 'also', 'proving', 'today',
'necessary', 'every', 'global', 'organization', 'reform', 'relevance', 'mind',
'yesterday', 'taken', 'historic', 'initiative', 'making', 'african', 'union',
'permanent', 'member', 'g20', 'similarly', 'also', 'mandate', 'multilateral',
'development', 'banks', 'decisions', 'direction', 'immediate', 'also', 'world',
'marked', 'rapid', 'change', 'require', 'transformation', 'also',
'sustainability', 'stability', 'come', 'let', 'us', 'pledge', 'take',
'resolutions', 'green', 'development', 'pact', 'action', 'plan', 'sdgs', 'high',
'level', 'principles', 'anticorruption', 'digital', 'public', 'infrastructure',
'mdb', 'reforms', 'fruition', 'highnesses', 'would', 'hear', 'thoughts',
'prime', 'ministers', 'remarks', 'closing', 'ceremony', 'g20', 'summit',
'complete', 'troika', 'pledge', 'unwavering', 'brazil', 'firmly', 'believe',
'leadership', 'g20', 'advance', 'common', 'objectives', 'president', 'brazil',
'lula', 'da', 'silva', 'pass', 'presidency', 'gavel', 'president', 'lula',
'thoughts', 'occasion', 'president', 'lulas', 'remarks', 'highnesses', 'aware',
'india', 'holds', 'g20', 'presidency', 'till', 'november', 'still', 'two',
'half', 'months', 'left', 'two', 'days', 'put', 'forth', 'many', 'things',
'given', 'suggestions', 'made', 'many', 'proposals', 'responsibility', 'look',
'suggestions', 'come', 'forward', 'see', 'thier', 'accelerated', 'propose',
'hold', 'another', 'virtual', 'session', 'g20', 'summit', 'end', 'november',
'session', 'review', 'topics', 'decided', 'summit', 'team', 'details',
'highnesses', 'words', 'officially', 'conclude', 'g20', 'summit', 'may', 'path',
'toward', 'one', 'earth', 'one', 'family', 'one', 'future', '  ', '  ',
'  ', 'means', 'let', 'entire', 'world', 'wellwishes', '140', 'crore',
'indians', 'express']
```

The term "FreqDist" typically refers to the Frequency Distribution function in natural language processing and text analysis. It helps in counting and analyzing the frequency of each unique element in a dataset, such as words in a text corpus

[29]:
```
# Create frequency distributions for positive, negative, and neutral words.
word_freq_positive = nltk.FreqDist(positive_words)
word_freq_negative = nltk.FreqDist(negative_words)
word_freq_neutral = nltk.FreqDist(neutral_words)
```

[30]:
```
# Print the frequency distribution of positive words.
print('The positive words frequency is:', word_freq_positive)
```

```
The positive words frequency is: <FreqDist with 103 samples and 218 outcomes>
```

[31]:
```
# Print the frequency distribution of negative words.
print('The negative words frequency is:', word_freq_negative)
```

```
The negative words frequency is: <FreqDist with 23 samples and 30 outcomes>
```

[32]:
```
# Print the frequency distribution of neutral words.
print('The neutral words frequency is:', word_freq_neutral)
```

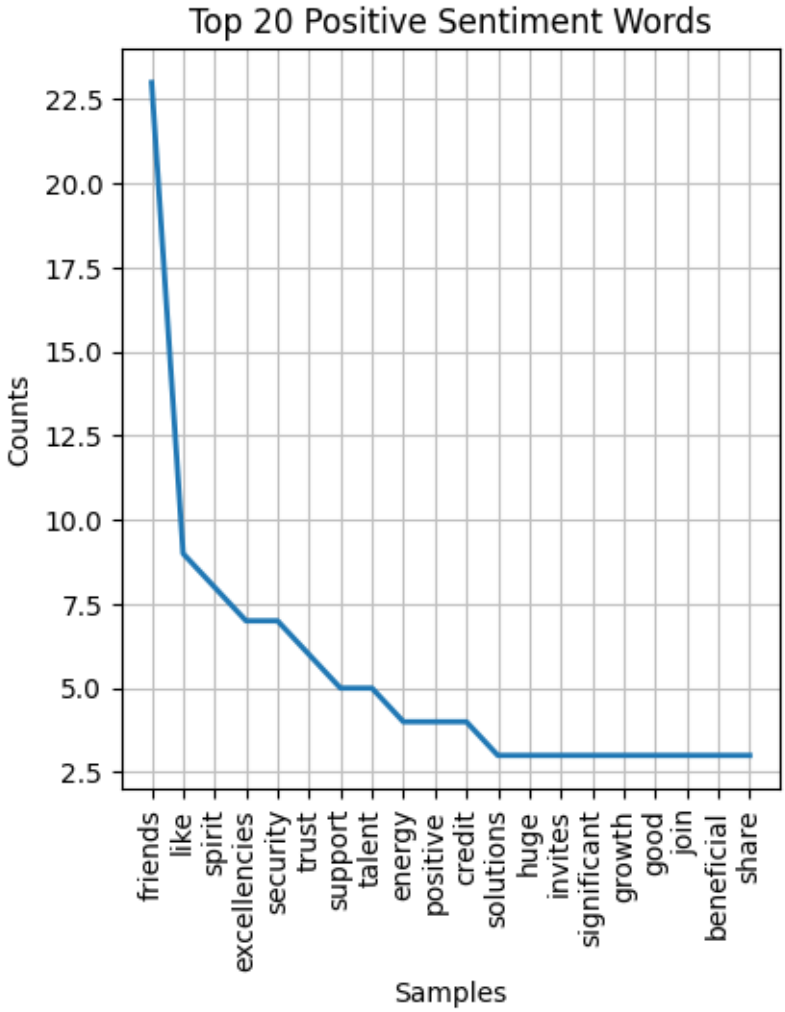
```
The neutral words frequency is: <FreqDist with 723 samples and 1479 outcomes>
```

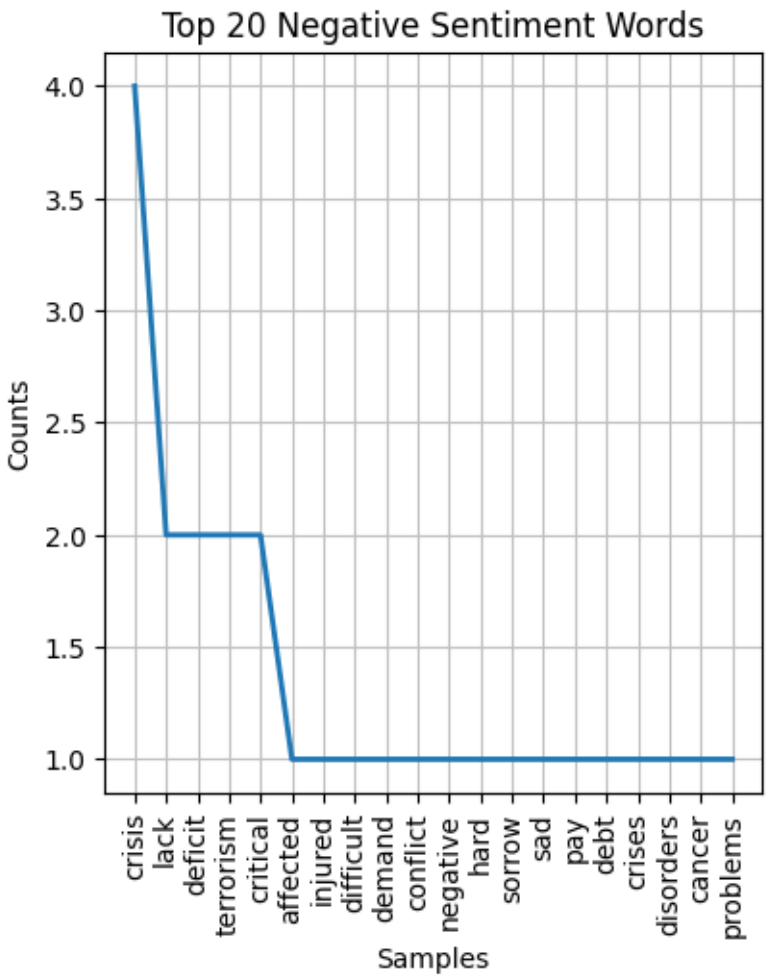
DATA VISUALISATION

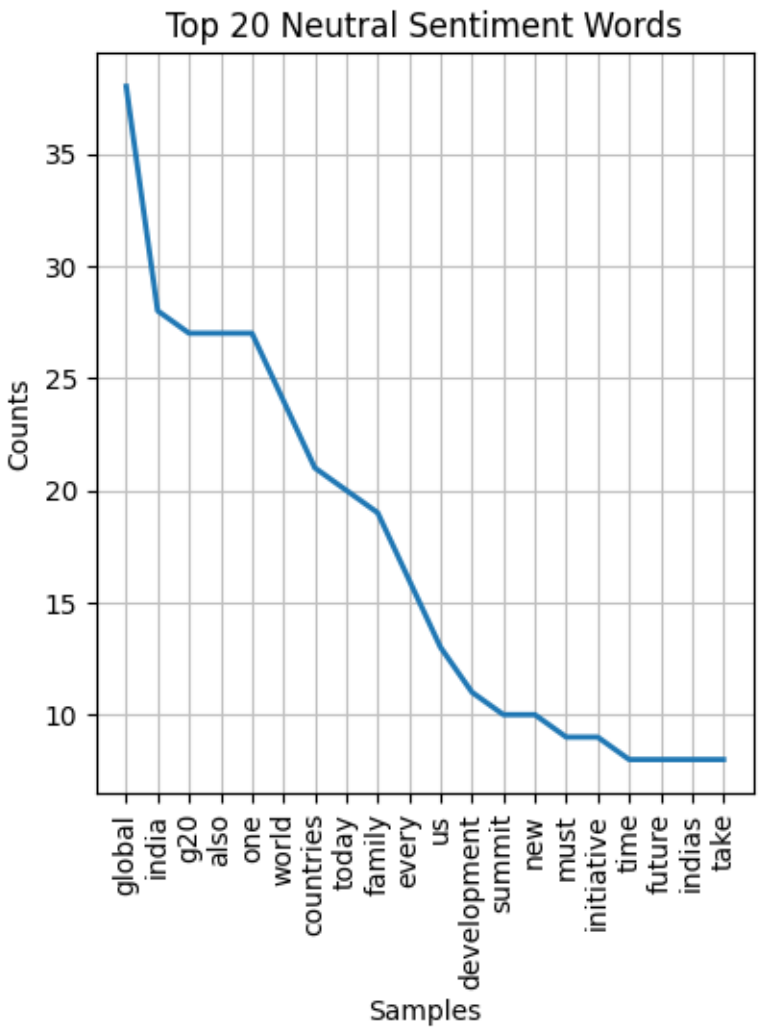
The plt.subplot function is used to position each subplot in a grid layout, allowing us to compare the word frequencies across different sentiments.

[33]:
```
# Create a new figure with a specific size (15 units in width, 5 units in␣
 ↪height).
plt.figure(figsize=(15, 5))

# Create the first subplot positioned in the first column (1) of a 1x3 grid.
plt.subplot(131)

# Plot the top 20 most frequent positive sentiment words and set a title.
word_freq_positive.plot(20, title="Top 20 Positive Sentiment Words")

# Create a new figure with the same size specification for the next subplot.
plt.figure(figsize=(15, 5))

# Create the second subplot positioned in the second column (2) of the same 1x3␣
 ↪grid.
plt.subplot(132)

# Plot the top 20 most frequent negative sentiment words and set a title.
word_freq_negative.plot(20, title="Top 20 Negative Sentiment Words")

# Create a new figure with the same size specification for the final subplot.
plt.figure(figsize=(15, 5))

# Create the third subplot positioned in the third column (3) of the same 1x3␣
 ↪grid.
plt.subplot(133)

# Plot the top 20 most frequent neutral sentiment words and set a title.
word_freq_neutral.plot(20, title="Top 20 Neutral Sentiment Words")

# Adjust the layout of subplots for better spacing.
plt.tight_layout()
```

```
# Display all subplots within the same figure.
plt.show()
```




Top 20 Negative Sentiment Words
Counts
4.0
3.5
3.0
2.5
2.0
1.5
1.0
crisis
lack
deficit
terrorism
critical
affected
injured
difficult
demand
conflict
negative
hard
sorrow
sad
pay
debt
crises
disorders
cancer
problems
Samples




```
<Figure size 640x480 with 0 Axes>
```

Creating New Dataframes

[34]:
```
# Create DataFrames for the top 20 most common positive sentiment words with␣
 ↪columns 'Word' and 'Frequency'.
df_positive = pd.DataFrame(word_freq_positive.most_common(20), columns=['Word',␣
 ↪'Frequency'])

# Create DataFrames for the top 20 most common negative sentiment words with␣
 ↪columns 'Word' and 'Frequency'.
df_negative = pd.DataFrame(word_freq_negative.most_common(20), columns=['Word',␣
 ↪'Frequency'])
```

```
# Create DataFrames for the top 20 most common neutral sentiment words with␣
 ↪columns 'Word' and 'Frequency'.
df_neutral = pd.DataFrame(word_freq_neutral.most_common(20), columns=['Word',␣
 ↪'Frequency'])
```

Plotting BAR GRaphs

[35]:
```
# Create bar charts for the top 20 positive sentiment words with 'Word' on the␣
 ↪x-axis and 'Frequency' on the y-axis.
fig_positive = px.bar(df_positive, x='Word', y='Frequency', title="Top 20␣
 ↪Positive Sentiment Words")

# Create bar charts for the top 20 negative sentiment words with 'Word' on the␣
 ↪x-axis and 'Frequency' on the y-axis.
fig_negative = px.bar(df_negative, x='Word', y='Frequency', title="Top 20␣
 ↪Negative Sentiment Words")

# Create bar charts for the top 20 neutral sentiment words with 'Word' on the␣
 ↪x-axis and 'Frequency' on the y-axis.
fig_neutral = px.bar(df_neutral, x='Word', y='Frequency', title="Top 20 Neutral␣
 ↪Sentiment Words")

# Display the bar chart for the top 20 positive sentiment words.
fig_positive.show()

# Display the bar chart for the top 20 negative sentiment words.
fig_negative.show()

# Display the bar chart for the top 20 neutral sentiment words.
fig_neutral.show()
```

PLOTTING WORD CLOUDS

[36]:
```
# Create word clouds for positive sentiment words with specific width, height,␣
 ↪and background color.
wordcloud_positive = WordCloud(width=800, height=400, background_color="white").
 ↪generate_from_frequencies(word_freq_positive)

# Create word clouds for negative sentiment words with specific width, height,␣
 ↪and background color.
wordcloud_negative = WordCloud(width=800, height=400, background_color="white").
 ↪generate_from_frequencies(word_freq_negative)

# Create word clouds for neutral sentiment words with specific width, height,␣
 ↪and background color.
wordcloud_neutral = WordCloud(width=800, height=400, background_color="white").
 ↪generate_from_frequencies(word_freq_neutral)
```

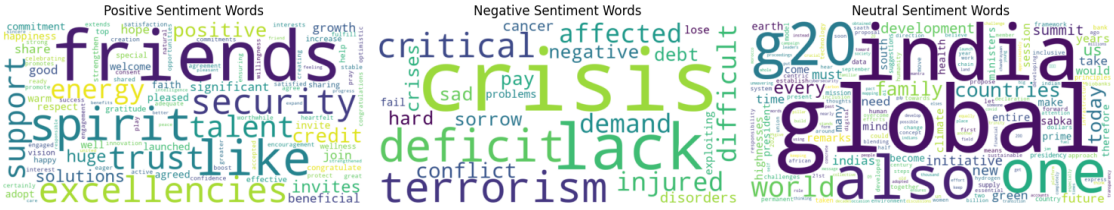
Bilinear Interpolation:

Bilinear Interpolation is used to display the word clouds with a smoother and more visually appealing appearance. Bilinear interpolation is a method that calculates the values of pixels in an image by considering their surrounding pixel values. In this context, it helps in producing a smoother and more continuous visual representation of the word cloud images.

Bilinear interpolation works by considering the values of nearby pixels and using a weighted average to estimate the value of a pixel at a non-integer coordinate. This interpolation method helps in reducing pixelation and jagged edges in the displayed image, resulting in a more visually pleasing and accurate representation.

[39]:
```
# Create a new figure with a specific size (15 units in width, 5 units in␣
 ↪height).
plt.figure(figsize=(15, 5))


# Create the first subplot positioned in the first column (1) of a 1x3 grid.
plt.subplot(131)
# Display the word cloud for positive sentiment words using bilinear␣
 ↪interpolation.
plt.imshow(wordcloud_positive, interpolation="bilinear")
# Turn off axis labels and ticks for a cleaner appearance.
plt.axis("off")
# Set the title for the first subplot.
plt.title("Positive Sentiment Words")



# Create the second subplot positioned in the second column (2) of the same 1x3␣
 ↪grid.
plt.subplot(132)
# Display the word cloud for negative sentiment words using bilinear␣
 ↪interpolation.
plt.imshow(wordcloud_negative, interpolation="bilinear")
# Turn off axis labels and ticks.
plt.axis("off")
# Set the title for the second subplot.
plt.title("Negative Sentiment Words")



# Create the third subplot positioned in the third column (3) of the same 1x3␣
 ↪grid.
plt.subplot(133)
# Display the word cloud for neutral sentiment words using bilinear␣
 ↪interpolation.
plt.imshow(wordcloud_neutral, interpolation="bilinear")
```

```
# Turn off axis labels and ticks.
plt.axis("off")
# Set the title for the third subplot.
plt.title("Neutral Sentiment Words")



# Adjust the layout of subplots for better spacing.
plt.tight_layout()
# Display all subplots within the same figure.
plt.show()
```



Individually Wordcloud

[40]:
```
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud_positive, interpolation="bilinear")
plt.title("Positive Sentiment Word Cloud")
plt.axis("off")
plt.show()
```

[41]:
```
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud_negative, interpolation="bilinear")
plt.title("Negative Sentiment Word Cloud")
plt.axis("off")
plt.show()
```

[42]:
```
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud_neutral, interpolation="bilinear")
plt.title("Neutral Sentiment Word Cloud")
plt.axis("off")
plt.show()
```

[43]:
```python
# Create a frequency distribution of lemmatized words.
word_freq = nltk.FreqDist(words_lemmatized)

# Generate a word cloud from the frequency distribution with specific width, height, and background color.
wordcloud = WordCloud(width=800, height=400, background_color="white").generate_from_frequencies(word_freq)
```

[45]:
```python
# Create and display a word cloud visualization of the lemmatized words from the speech, using bilinear interpolation for smoother visuals.
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud of Speech")
plt.show()
```

Percentage Summarization

[46]:
```
total_words = len(words_filtered)
positive_percentage = (len(positive_words) / total_words) * 100
negative_percentage = (len(negative_words) / total_words) * 100
neutral_percentage = (len(neutral_words) / total_words) * 100

print("Positive Sentiment Percentage:", positive_percentage)
print("Negative Sentiment Percentage:", negative_percentage)
print("Neutral Sentiment Percentage:", neutral_percentage)
```

```
Positive Sentiment Percentage: 12.623045744064854
Negative Sentiment Percentage: 1.7371163867979156
Neutral Sentiment Percentage: 85.63983786913724
```

Creating new Data

[47]:
```
data = {"Sentiment" : ['Positive' , 'Negative' , 'Neutral'] ,
        "Percentage" : [positive_percentage , negative_percentage ,␣
 ↪neutral_percentage]
        }
```

Dataframe

[48]:
```
df_percentages = pd.DataFrame(data)
```

[49]:
```
df_percentages
```

[49]:

	Sentiment	Percentage
0	Positive	12.623046
1	Negative	1.737116
2	Neutral	85.639838

Plotting Overall Graph

[51]:
```
fig = px.bar(df_percentages, x='Sentiment', y='Percentage', color='Sentiment',
             labels={'Sentiment': 'Sentiment Category', 'Percentage':␣
 ↪'Percentage (%)'},
             title='Percentage of Words in Each Sentiment Category')
fig.show()
```

Gensim

Gensim is an open-source natural language processing library in Python.

It specializes in topic modeling, document similarity analysis, and other text processing tasks. Gensim provides tools for training and using word embeddings, such as Word2Vec, and is widely used for text analysis and machine learning applications.

Latent Dirichlet Allocation (LDA) models:

Latent Dirichlet Allocation (LDA) is a statistical model used for topic modeling in natural language processing. It assumes that documents are mixtures of topics, and topics are mixtures of words. LDA helps uncover hidden topic structures within a collection of documents and is widely applied in text analysis, document clustering, and recommendation systems.

[53]:
```
import gensim
from gensim import corpora
from gensim.models.ldamodel import LdaModel
```

[54]:
```
# Create a dictionary from a list of preprocessed words (words_filtered).
dictionary = corpora.Dictionary([words_filtered])
```

[55]:
```
# Create a corpus representation using the dictionary by converting the␣
 ↪preprocessed words (words_filtered) into a bag-of-words format.
corpus = [dictionary.doc2bow(words_filtered)]
```

[56]:
```
# Create an LDA (Latent Dirichlet Allocation) model by training it on the␣
 ↪corpus.
lda_model = LdaModel(corpus, num_topics=5, id2word=dictionary, passes=15)

# Extract and print the top words for each of the identified topics in the LDA␣
 ↪model.
topics = lda_model.print_topics(num_words=5)
for topic in topics:
    print(topic)
```

```
(0, '0.001*"global" + 0.001*"one" + 0.001*"india" + 0.001*"g20" +
0.001*"world"')
(1, '0.001*"global" + 0.001*"also" + 0.001*"one" + 0.001*"india" +
0.001*"friends"')
(2, '0.001*"india" + 0.001*"global" + 0.001*"also" + 0.001*"g20" +
0.001*"countries"')
(3, '0.020*"global" + 0.015*"india" + 0.014*"g20" + 0.014*"one" + 0.014*"also"')
(4, '0.001*"global" + 0.001*"also" + 0.001*"india" + 0.001*"one" +
0.001*"world"')
```

[57]:
```python
# Import the spaCy library and load the English language model.
import spacy
nlp = spacy.load("en_core_web_sm")

# Process the text using spaCy to identify named entities.
doc = nlp(speech_text)

# Extract and print the named entities along with their labels.
entities = [(ent.text, ent.label_) for ent in doc.ents]

for entity, label in entities:
    print(f"Entity: {entity}, Label: {label}")
```

```
Entity: September 08, 2023, Label: DATE
Entity: Excellencies, Label: ORG
Entity: Namaskar, Label: PERSON
Entity: Morocco, Label: GPE
Entity: a moment ago, Label: TIME
Entity: Morocco, Label: GPE
Entity: Excellencies, Label: ORG
Entity: G-20, Label: GPE
Entity: India, Label: GPE
Entity: today, Label: DATE
Entity: just a few kilometers, Label: TIME
Entity: nearly two and a half thousand years old, Label: CARDINAL
Entity: Prakrit, Label: GPE
Entity:    , Label: ORG
Entity: Two and a half thousand years ago, Label: DATE
Entity: India, Label: GPE
Entity: The 21st century, Label: DATE
Entity: years, Label: DATE
Entity: Centric, Label: PERSON
Entity: COVID, Label: ORG
Entity: Today, Label: DATE
Entity: G-20, Label: GPE
Entity: India, Label: GPE
Entity: first, Label: ORDINAL
Entity: Global Trust Deficit, Label: ORG
```

Entity: Sabka Saath, Label: PERSON
Entity: Sabka Vikas, Label: PERSON
Entity: Sabka Vishwas, Label: PERSON
Entity: Sabka Prayas', Label: ORG
Entity: East, Label: LOC
Entity: West, Label: LOC
Entity: Management of Food, Fuel, Label: ORG
Entity: Fertilizer, Label: PERSON
Entity: Terrorism and Cybersecurity, Label: ORG
Entity: Health, Energy, Label: ORG
Entity: Water Security, Label: PERSON
Entity: India, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: Sabka Saath', Label: PERSON
Entity: Millions, Label: CARDINAL
Entity: Indians, Label: NORP
Entity: Over 200, Label: CARDINAL
Entity: more than 60, Label: CARDINAL
Entity: Sabka Saath', Label: PERSON
Entity: India, Label: GPE
Entity: the African Union, Label: ORG
Entity: G-20, Label: PRODUCT
Entity: the Chairperson of the African Union, Label: WORK_OF_ART
Entity: G-20, Label: PRODUCT
Entity: the G20 Summit Session, Label: FAC
Entity: 1
September 09, 2023, Label: DATE
Entity: India, Label: GPE
Entity: democratic, Label: NORP
Entity: Vasudhaiva Kutumbakam, Label: PERSON
Entity: Indian, Label: NORP
Entity: One, Label: CARDINAL
Entity: Earth, Label: LOC
Entity: One, Label: CARDINAL
Entity: Earth, Label: LOC
Entity: India, Label: GPE
Entity: India, Label: GPE
Entity: the 'International Year of Millets', Label: DATE
Entity: this year, Label: DATE
Entity: Climate Security, Label: ORG
Entity: India, Label: GPE
Entity: One, Label: CARDINAL
Entity: One, Label: CARDINAL
Entity: Today, Label: DATE
Entity: India, Label: GPE
Entity: Millions, Label: CARDINAL
Entity: Indian, Label: NORP
Entity: Earth, Label: LOC

Entity: the 'National Green Hydrogen Mission', Label: ORG
Entity: India, Label: GPE
Entity: India, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: Global Hydrogen Ecosystem, Label: ORG
Entity: Friends, Label: ORG
Entity: the 21st century, Label: DATE
Entity: Trillions of dollars, Label: MONEY
Entity: Along, Label: GPE
Entity: India, Label: GPE
Entity: the Global South, Label: LOC
Entity: this year, Label: DATE
Entity: 2023, Label: DATE
Entity: 100 billion dollars, Label: MONEY
Entity: first, Label: ORDINAL
Entity: G-20, Label: PRODUCT
Entity: Friends, Label: ORG
Entity: today, Label: DATE
Entity: India, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: Today, Label: DATE
Entity: the hour, Label: TIME
Entity: up to 20 percent, Label: PERCENT
Entity: today, Label: DATE
Entity: the Global Biofuel Alliance, Label: ORG
Entity: India, Label: GPE
Entity: Friends, Label: ORG
Entity: Carbon Credit, Label: ORG
Entity: decades, Label: DATE
Entity: Carbon Credit, Label: ORG
Entity: Green Credit, Label: ORG
Entity: G-20, Label: PRODUCT
Entity: Friends, Label: ORG
Entity: India, Label: GPE
Entity: Moon Mission, Chandrayaan, Label: ORG
Entity: India, Label: GPE
Entity: the 'G20 Satellite Mission for Environment and Climate Observation,
Label: ORG
Entity: the Global South, Label: LOC
Entity: India, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: Friends, Label: ORG
Entity: the G20 Summit Session, Label: FAC
Entity: 2
September 09, 2023, Label: DATE
Entity: the New Delhi G20 Leaders' Summit Declaration, Label: ORG
Entity: Sherpa, Label: PERSON
Entity: Vedas, Label: ORG

```
Entity: thousands of years ago, Label: DATE
Entity: One, Label: CARDINAL
Entity: One, Label: CARDINAL
Entity: India, Label: GPE
Entity: India, Label: GPE
Entity: India, Label: GPE
Entity: Aadhar, Label: GPE
Entity: The World Bank, Label: ORG
Entity: Trinity, Label: ORG
Entity: just 6 years, Label: DATE
Entity: 47 years, Label: DATE
Entity: India, Label: GPE
Entity: 360 billion dollars, Label: MONEY
Entity: the past decade, Label: DATE
Entity: approximately 33 billion dollars, Label: MONEY
Entity: nearly 1.25%, Label: PERCENT
Entity: the Global South, Label: LOC
Entity: India, Label: GPE
Entity: One, Label: CARDINAL
Entity: Global Skill Mapping, Label: WORK_OF_ART
Entity: the Global South, Label: LOC
Entity: Covid, Label: PERSON
Entity: over decades, Label: DATE
Entity: One, Label: CARDINAL
Entity: today, Label: DATE
Entity: India, Label: GPE
Entity: One, Label: CARDINAL
Entity: Family, Label: ORG
Entity: the "Action Plan to Accelerate Sustainable Development Goals, Label: LAW
Entity: One, Label: CARDINAL
Entity: Wellness, Label: ORG
Entity: The WHO Global Centre for Traditional Medicine, Label: ORG
Entity: India, Label: GPE
Entity: Today, Label: DATE
Entity: India, Label: GPE
Entity: Approximately 45%, Label: PERCENT
Entity: STEM (Science, Technology, Label: ORG
Entity: Mathematics, Label: NORP
Entity: India, Label: GPE
Entity: Today, Label: DATE
Entity: India, Label: GPE
Entity: Today, Label: DATE
Entity: around 90 million, Label: CARDINAL
Entity: Indian, Label: NORP
Entity: the Self Help Groups, Label: ORG
Entity: the 21st century, Label: DATE
Entity: three, Label: CARDINAL
Entity: First, Label: ORDINAL
```

```
Entity: 5%, Label: PERCENT
Entity: the Global South, Label: LOC
Entity: Second, Label: ORDINAL
Entity: G20 Talent Visa, Label: WORK_OF_ART
Entity: Third, Label: ORDINAL
Entity: Global Bio, Label: ORG
Entity: WHO, Label: ORG
Entity: Global Bio, Label: PERSON
Entity: India, Label: GPE
Entity: the G20 Summit Session, Label: FAC
Entity: 3, Label: CARDINAL
Entity: Excellencies, Label: ORG
Entity: Yesterday, Label: DATE
Entity: One, Label: CARDINAL
Entity: Earth, Label: LOC
Entity: One, Label: CARDINAL
Entity: today, Label: DATE
Entity: G-20, Label: PRODUCT
Entity: One, Label: CARDINAL
Entity: Earth, Label: LOC
Entity: One, Label: CARDINAL
Entity: Global Village, Label: GPE
Entity: Human Centric Vision, Label: ORG
Entity: Centric Approach, Label: PERSON
Entity: Today, Label: DATE
Entity: India, Label: GPE
Entity: India, Label: GPE
Entity: Chandrayaan Mission, Label: PERSON
Entity: India, Label: GPE
Entity: last-mile, Label: QUANTITY
Entity: India, Label: GPE
Entity: Digital Public Infrastructure, Label: ORG
Entity: G20 Principles on Harnessing Data for Development, Label: WORK_OF_ART
Entity: Data for Development Capacity Building Initiative, Label: ORG
Entity: Global South, Label: LOC
Entity: India, Label: GPE
Entity: Today, Label: DATE
Entity: Artificial Intelligence, Label: ORG
Entity: 2019, Label: DATE
Entity: AI, Label: ORG
Entity: Today, Label: DATE
Entity: one, Label: CARDINAL
Entity: India, Label: GPE
Entity: AI, Label: ORG
Entity: Socio-Economic Development, Label: ORG
Entity: Friends, Label: WORK_OF_ART
Entity: Today, Label: DATE
Entity: i.e, Label: NORP
```

```
Entity: Basel, Label: GPE
Entity: Today, Label: DATE
Entity: the "United Nations Security Council, Label: ORG
Entity: UN, Label: ORG
Entity: today, Label: DATE
Entity: 51, Label: CARDINAL
Entity: UN, Label: ORG
Entity: Today, Label: DATE
Entity: UN, Label: ORG
Entity: around 200, Label: CARDINAL
Entity: UNSC, Label: ORG
Entity: today, Label: DATE
Entity: the past years, Label: DATE
Entity: Today, Label: DATE
Entity: yesterday, Label: DATE
Entity: the African Union, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: Multilateral Development Banks, Label: ORG
Entity: Green Development Pact, Label: ORG
Entity: Digital Public Infrastructure, Label: ORG
Entity: Excellencies, Label: ORG
Entity: G20, Label: ORG
Entity: Friends, Label: ORG
Entity: Troika, Label: ORG
Entity: Brazil, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: Brazil, Label: GPE
Entity: Lula da Silva, Label: PERSON
Entity: Lula, Label: PERSON
Entity: Lula, Label: PERSON
Entity: Excellencies, Label: ORG
Entity: India, Label: GPE
Entity: G-20, Label: PRODUCT
Entity: November, Label: DATE
Entity: two and a half months, Label: DATE
Entity: these two days, Label: DATE
Entity: the G-20 Summit, Label: FAC
Entity: the end of November, Label: DATE
Entity: Excellencies, Label: ORG
Entity: this G-20 Summit, Label: FAC
Entity: May, Label: DATE
Entity: One, Label: CARDINAL
Entity: Earth, Label: LOC
Entity: One, Label: CARDINAL
Entity:    , Label: ORG
Entity: 140, Label: CARDINAL
Entity: Indians, Label: NORP
```

KeyBERT is a Python library for keyword extraction and generation. It leverages transformer-based language models like BERT to automatically identify important keywords or phrases from text, aiding in summarization and content analysis tasks.

These keywords in output are representative of the most important terms or phrases in the text, as determined by the KeyBERT model.

[58]:
```
# Import the KeyBERT library for keyword extraction.
from keybert import KeyBERT

# Initialize the KeyBERT keyword extractor.
kw_extractor = KeyBERT()

# Extract keywords from the speech_text using the KeyBERT model.
keywords = kw_extractor.extract_keywords(speech_text)

# Iterate through the extracted keywords and print them.
for keyword in keywords:
    print(keyword[0])
```

```
Downloading (…)e9125/.gitattributes:   0%|          | 0.00/1.18k [00:00<?, ?B/s]
Downloading (…)_Pooling/config.json:   0%|          | 0.00/190 [00:00<?, ?B/s]
Downloading (…)7e55de9125/README.md:   0%|          | 0.00/10.6k [00:00<?, ?B/s]
Downloading (…)55de9125/config.json:   0%|          | 0.00/612 [00:00<?, ?B/s]
Downloading (…)ce_transformers.json:   0%|          | 0.00/116 [00:00<?, ?B/s]
Downloading (…)125/data_config.json:   0%|          | 0.00/39.3k [00:00<?, ?B/s]
Downloading pytorch_model.bin:   0%|          | 0.00/90.9M [00:00<?, ?B/s]
Downloading (…)nce_bert_config.json:   0%|          | 0.00/53.0 [00:00<?, ?B/s]
Downloading (…)cial_tokens_map.json:   0%|          | 0.00/112 [00:00<?, ?B/s]
Downloading (…)e9125/tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]
Downloading (…)okenizer_config.json:   0%|          | 0.00/350 [00:00<?, ?B/s]
Downloading (…)9125/train_script.py:   0%|          | 0.00/13.2k [00:00<?, ?B/s]
Downloading (…)7e55de9125/vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]
Downloading (…)5de9125/modules.json:   0%|          | 0.00/349 [00:00<?, ?B/s]
pillar
mantra
summit

minister
```

NRClex:

NRClex, short for National Research Council Emotion Lexicon, is a lexical resource that provides a comprehensive list of words and their associated emotions. It categorizes words based on emotions such as anger, joy, sadness, and fear, enabling sentiment analysis and emotion detection in text data. Researchers and analysts use NRClex to understand the emotional tone of written content and extract insights about sentiment and emotional context.

[59]:
```
# Import the NRCLex library for emotion analysis.
from nrclex import NRCLex

# Initialize an NRCLex object with the cleaned speech text.
text_emotion = NRCLex(speech_text_cleaned)

# Calculate the frequency of various emotions in the text.
emotions = text_emotion.affect_frequencies

# Iterate through the emotions and their frequencies, then print them.
for emotion, frequency in emotions.items():
    print(f"Emotion: {emotion}, Frequency: {frequency}")
```

```
Emotion: fear, Frequency: 0.0359375
Emotion: anger, Frequency: 0.0265625
Emotion: anticip, Frequency: 0.0
Emotion: trust, Frequency: 0.203125
Emotion: surprise, Frequency: 0.04375
Emotion: positive, Frequency: 0.36875
Emotion: negative, Frequency: 0.05625
Emotion: sadness, Frequency: 0.0234375
Emotion: disgust, Frequency: 0.0109375
Emotion: joy, Frequency: 0.1125
Emotion: anticipation, Frequency: 0.11875
```

textstat

Textstat is a Python library that helps analyze the readability and complexity of text. It provides functions to calculate readability scores, word and sentence counts, and other linguistic statistics. Textstat is useful for assessing the readability of documents, aiding in content creation, and tailoring text for specific audiences.

[61]:
```
# Import the textstat library for text readability analysis.
import textstat

# Calculate the Flesch Reading Ease score for the cleaned speech text.
flesch_score = textstat.flesch_reading_ease(speech_text_cleaned)

# Calculate the Flesch-Kincaid Grade Level for the cleaned speech text.
flesch_grade = textstat.flesch_kincaid_grade(speech_text_cleaned)

# Calculate the SMOG Index for the cleaned speech text.
smog_index = textstat.smog_index(speech_text_cleaned)
```

```
# Print the calculated readability scores.
print(f"Flesch Reading Ease Score: {flesch_score}")
print(f"Flesch-Kincaid Grade Level: {flesch_grade}")
print(f"SMOG Index: {smog_index}")
```

```
Flesch Reading Ease Score: -3052.7
Flesch-Kincaid Grade Level: 1203.7
SMOG Index: 0.0
```

NLTK

NLTK (Natural Language Toolkit) is a Python library widely used for natural language processing and text analysis. It provides tools and resources for tasks like tokenization, stemming, tagging, parsing, and sentiment analysis. NLTK is valuable for researchers, developers, and educators working with text data, offering a comprehensive set of functionalities and linguistic resources.

Pointwise Mutual Information (PMI) is a measure used in natural language processing to assess the statistical association between two words in a text corpus. It quantifies how much more often two words co-occur together than if they were independent, indicating their degree of association or relevance in a specific context.

[62]:
```
# Import necessary components from the NLTK library for bigram collocation␣
 ↪analysis.
from nltk.collocations import BigramAssocMeasures, BigramCollocationFinder

# Tokenize the cleaned speech text into words.
tokens = nltk.word_tokenize(speech_text_cleaned)

# Initialize BigramAssocMeasures to use Pointwise Mutual Information (PMI) as␣
 ↪the scoring measure.
bigram_measures = BigramAssocMeasures()

# Create a BigramCollocationFinder to find bigram collocations from the␣
 ↪tokenized words.
finder = BigramCollocationFinder.from_words(tokens)

# Score and rank bigram collocations based on PMI (Pointwise Mutual␣
 ↪Information).
pmi_scores = finder.score_ngrams(bigram_measures.pmi)

# Print the top 10 bigram collocations along with their PMI scores.
for bigram, pmi in pmi_scores[:10]:
    print(f"Bigram: {bigram}, PMI: {pmi}")
```

```
Bigram: ('140', 'crore'), PMI: 11.588246152044817
Bigram: ('51', 'founding'), PMI: 11.588246152044817
Bigram: ('60', 'cities'), PMI: 11.588246152044817
Bigram: ('90', 'million'), PMI: 11.588246152044817
```

```
Bigram: ('aadhar', 'identity'), PMI: 11.588246152044817
Bigram: ('after', 'covid19'), PMI: 11.588246152044817
Bigram: ('ancient', 'vedas'), PMI: 11.588246152044817
Bigram: ('anemia', 'endocrine'), PMI: 11.588246152044817
Bigram: ('artificial', 'intelligence'), PMI: 11.588246152044817
Bigram: ('between', 'east'), PMI: 11.588246152044817
```