Twitter Data Analysis for Stemming and Lemmatization

In [ ]:
```
import http.client
import json
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer
from nltk.stem import WordNetLemmatizer
import spacy
```

Initialize stemmers and lemmatizers

In [ ]:
```
porter = PorterStemmer()
snowball = SnowballStemmer("english")
lancaster = LancasterStemmer()
wordnet_lemmatizer = WordNetLemmatizer()
spacy_nlp = spacy.load("en_core_web_sm")
```

Define stemming and lemmatization functions

In [ ]:
```
nlp = spacy.load("en_core_web_sm")
def porter_stemmer(text):
    tokens = word_tokenize(text)
    stemmed = [porter.stem(token) for token in tokens]
    return ' '.join(stemmed)

def snowball_stemmer(text):
    tokens = word_tokenize(text)
    stemmed = [snowball.stem(token) for token in tokens]
    return ' '.join(stemmed)

def lancaster_stemmer(text):
    tokens = word_tokenize(text)
    stemmed = [lancaster.stem(token) for token in tokens]
    return ' '.join(stemmed)

def wordnet_lemmatizer(text):
    wordnet = WordNetLemmatizer()
    tokens = word_tokenize(text)
    lemmatized = [wordnet.lemmatize(token) for token in tokens]
    return ' '.join(lemmatized)

def spacy_lemmatizer(text):
    doc = nlp(text)
    lemmatized = [token.lemma_ for token in doc]
    return ' '.join(lemmatized)
```

Function to fetch tweets using RapidAPI

In [ ]:
```
def fetch_tweets(query, min_retweets=20, min_likes=20, limit=5, start_date="2022-01-01", lang
    conn = http.client.HTTPSConnection("twitter154.p.rapidapi.com")
    headers = {
        'x-rapidapi-key': "5c75308bf6msh2b302bcfa9a8e0ep109e11jsn949c33074b98",
        'x-rapidapi-host': "twitter154.p.rapidapi.com"
    }

    endpoint = f"/search/search/continuation?query={query}&section=top&min_retweets={min_retw
    conn.request("GET", endpoint, headers=headers)
    res = conn.getresponse()

    if res.status != 200:
        raise Exception(f"API request failed with status {res.status}")

    data = res.read()
    return json.loads(data)
```

Converting Data from API response to Dataframe

In [ ]:
```
# Fetch tweets containing the query indian Tourism
tweets_data = fetch_tweets("incredibleindia")

# Extract the tweet texts
tweets = [tweet['text'] for tweet in tweets_data.get('results', [])]

# Check if tweets are retrieved
if not tweets:
    raise Exception("No tweets were retrieved from the API")

# Convert to DataFrame for convenience
tweets_df = pd.DataFrame(tweets, columns=['Tweet'])
```

Adding New Columns for Stemmed and Lemmatized Text

In [ ]:
```
# Apply stemming and lemmatization functions to the DataFrame
tweets_df['Porter Stemmed'] = tweets_df['Tweet'].apply(porter_stemmer)
tweets_df['Snowball Stemmed'] = tweets_df['Tweet'].apply(snowball_stemmer)
tweets_df['Lancaster Stemmed'] = tweets_df['Tweet'].apply(lancaster_stemmer)
tweets_df['WordNet Lemmatized'] = tweets_df['Tweet'].apply(wordnet_lemmatizer)
tweets_df['Spacy Lemmatized'] = tweets_df['Tweet'].apply(spacy_lemmatizer)

# Display the DataFrame
tweets_df
```

Out[ ]:

	Tweet	Porter Stemmed	Snowball Stemmed	Lancaster Stemmed	WordNet Lemmatized	Spacy Lemmatized
0	Snehatheeram Beach, Kerala, India.\n\n[viji....	snehatheeram beach , kerala , india . [viji...	snehatheeram beach , kerala , india . [viji...	snehatheeram beach , keral , ind . [viji.t....	Snehatheeram Beach , Kerala , India . [viji...	Snehatheeram Beach , Kerala , India . \n\n [...
1	When I arrived in #India nearly 4 years ago, I...	when i arriv in # india nearli 4 year ago , i ...	when i arriv in # india near 4 year ago , i ha...	when i ar in # ind near 4 year ago , i had man...	When I arrived in # India nearly 4 year ago , ...	when I arrive in # India nearly 4 year ago , I...
2	Continuing to discover #IncredibleIndia 😍\nToo...	continu to discov # incredibleindia 😍 took my ...	continu to discov # incredibleindia 😍 took my ...	continu to discov # incredibleind 😍 took my fa...	Continuing to discover # IncredibleIndia 😍 Too...	continue to discover # IncredibleIndia 😍 \n ta...
3	Feeling blessed my siblings came to visit me i...	feel bless my sibl came to visit me in # incre...	feel bless my sibl came to visit me in # incre...	feel bless my sibl cam to visit me in # incred...	Feeling blessed my sibling came to visit me in...	feel bless my sibling come to visit I in # Inc...

Saving Data To A CSV file for Ease of Access

In [ ]:
```
# Save the DataFrame to a CSV file
tweets_df.to_csv('stemming_lemmatization_comparison.csv', index=False)
```

Conclusion and Observations

In this analysis, we explored various methods for stemming and lemmatization on a dataset of tweets related to travel and experiences in India. The goal was to understand how different preprocessing techniques affect the text data and to compare the outputs generated by each method.

Stemming and Lemmatization Methods

1. **Porter Stemmer**: A classic and widely-used stemming algorithm that reduces words to their root forms.
2. **Snowball Stemmer**: An improvement over the Porter Stemmer, known for being more aggressive in its approach.
3. **Lancaster Stemmer**: The most aggressive stemmer among the three, often resulting in shorter stems.
4. **WordNet Lemmatizer**: Uses a dictionary-based approach to reduce words to their base or dictionary form.
5. **Spacy Lemmatizer**: Utilizes Spacy's natural language processing capabilities to lemmatize words based on context.

Observations

Porter Stemmer

- **Output**: "snehatheeram beach , kerala , india . [🇮🇳 viji..."; "when i arriv in # india nearli 4 year ago , i ..."; "continu to discov # incredibleindia 😍 took my ..."; "feel bless my sibl came to visit me in # incre..."
- **Comments**: The Porter Stemmer effectively reduced words to their root forms, but it sometimes produced non-intuitive stems (e.g., "arriv" instead of "arrive").

Snowball Stemmer

- **Output**: "snehatheeram beach , kerala , india . [🇮🇳 viji..."; "when i arriv in # india near 4 year ago , i ha..."; "continu to discov # incredibleindia 😍 took my ..."; "feel bless my sibl came to visit me in # incre..."
- **Comments**: The Snowball Stemmer produced results similar to the Porter Stemmer but was slightly more aggressive in certain cases, such as "nearli" becoming "near."

Lancaster Stemmer

- **Output**: "snehatheeram beach , keral , ind . [🇮🇳 viji.t..."; "when i ar in # ind near 4 year ago , i had man..."; "continu to discov # incredibleind 😍 took my fa..."; "feel bless my sibl cam to visit me in # incred..."
- **Comments**: The Lancaster Stemmer was the most aggressive, often resulting in very short and sometimes confusing stems (e.g., "keral" instead of "kerala").

WordNet Lemmatizer

- **Output**: "Snehatheeram Beach , Kerala , India . [🇮🇳 viji..."; "When I arrived in # India nearly 4 year ago , ..."; "Continuing to discover # IncredibleIndia 😍 Too..."; "Feeling blessed my sibling came to visit me in..."
- **Comments**: The WordNet Lemmatizer provided more contextually accurate results, preserving the meaning of the original words better than the stemmers.

Spacy Lemmatizer

- **Output**: "Snehatheeram Beach , Kerala , India . \n\n [🇮🇳 ..."; "when I arrive in # India nearly 4 year ago , I..."; "continue to discover # IncredibleIndia 😍 \n ta..."; "feel bless my sibling come to visit I in # Inc..."
- **Comments**: Similar to the WordNet Lemmatizer, the Spacy Lemmatizer produced contextually accurate and readable results, with slight differences in handling punctuation and spacing.

Conclusion

- **Stemmers** (Porter, Snowball, Lancaster) are useful for reducing words to their root forms, which can be beneficial for certain text processing tasks where the exact meaning of words is not as important.
- **Lemmatizers** (WordNet, Spacy) are more sophisticated and provide contextually accurate base forms of words, preserving their meanings and making the text more readable and understandable.

- The choice between stemming and lemmatization should be based on the specific requirements of the text processing task. For tasks requiring more accurate and readable text, lemmatization is preferred. For simpler, more computationally efficient tasks, stemming may suffice.