Codeboosters Tech

team_codeboosters

www.codeboosters.in

🧠 Topic 1: Natural Language Processing (NLP) - Introduction

Theory:
Natural Language Processing (NLP) is a field of AI that focuses on the interaction between computers and human languages. It enables machines to read, understand, and derive meaning from human language.

Use Cases:

- Chatbots (like this one!)
- Language translation
- Sentiment analysis
- Speech recognition

📘 1. NLTK (Natural Language Toolkit)

Theory:
NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources.

✅ Example 1: Tokenizing South Indian food names

python

```python
import nltk
nltk.download('punkt')  # Downloads tokenizer models
```

```
from nltk.tokenize import word_tokenize

text = "Dosa Idli Vada Pongal Upma"
tokens = word_tokenize(text)
print("Tokens:", tokens)
```

Explanation:

- `word_tokenize` breaks the sentence into words.
- Input: `"Dosa Idli Vada Pongal Upma"`
- Output: `['Dosa', 'Idli', 'Vada', 'Pongal', 'Upma']`

✅ Example 2: Tokenizing IPL Team names

python

```
text = "Chennai Super Kings, Royal Challengers Bangalore, Mumbai Indians"
tokens = word_tokenize(text)
print("Tokens:", tokens)
```

Output: `['Chennai', 'Super', 'Kings', ',', 'Royal', 'Challengers', 'Bangalore', ',', 'Mumbai', 'Indians']`

📘 2. TextBlob

Theory:
TextBlob is a simple NLP library for Python that provides a consistent API for diving into common NLP tasks like part-of-speech tagging, noun phrase extraction, and sentiment analysis.

python

```
!pip install -q textblob
from textblob import TextBlob
```

```
text = TextBlob("Idli is delicious and Dosa is amazing")
print("Sentiment:", text.sentiment)
```

Explanation:

- `TextBlob` automatically evaluates sentiment (`polarity` and `subjectivity`)
- Polarity: [-1, 1] — negative to positive
- Subjectivity: [0, 1] — fact to opinion

✅ Example 2: IPL Team commentary sentiment

python

```
comment = TextBlob("Chennai Super Kings played exceptionally well today!")
print("Sentiment:", comment.sentiment)
```

📘 3. Tokenization

Theory:
Tokenization is the process of breaking down text into smaller units called tokens. These could be words, characters, or subwords.

✅ Example 1: Word Tokenization (South Indian foods)

python

```
text = "I love Pongal and Vada for breakfast"
tokens = word_tokenize(text)
print("Word Tokens:", tokens)
```

✅ Example 2: Sentence Tokenization (IPL context)

python

```python
from nltk.tokenize import sent_tokenize

text = "Chennai Super Kings won the match. Mumbai Indians gave a tough fight."
sentences = sent_tokenize(text)
print("Sentences:", sentences)
```

📘 4. Stemming and Lemmatization

Theory:

- **Stemming:** Reduces words to their root form (e.g., "eating" → "eat")
- **Lemmatization:** Smarter than stemming, reduces words to their base form using dictionary knowledge

✅ Example 1: Stemming food review

python

```python
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()
words = ["eating", "eaten", "eats", "loving"]
stems = [stemmer.stem(word) for word in words]
print("Stemmed Words:", stems)
```

✅ Example 2: Lemmatization on IPL match description

python

```python
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
words = ["playing", "played", "runs"]
```

```
lemmas = [lemmatizer.lemmatize(word, pos='v') for word in words]
print("Lemmatized Words:", lemmas)
```

📘 5. Stop Words

Theory:
Stop words are common words (like "is", "and", "the") that are usually removed from text before processing.

✅ Example 1: Removing stop words from South Indian menu

python

```
from nltk.corpus import stopwords
nltk.download('stopwords')

words = word_tokenize("I like Idli and Dosa with chutney")
filtered = [w for w in words if w.lower() not in stopwords.words('english')]
print("Filtered Words:", filtered)
```

✅ Example 2: IPL Commentary

python

```
sentence = "The Royal Challengers Bangalore are playing very well today"
filtered = [w for w in word_tokenize(sentence) if w.lower() not in
stopwords.words('english')]
print("Filtered Commentary:", filtered)
```

📘 6. Visualizing Word Frequency using Pandas and WordCloud

python

```python
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt

text = "Dosa Dosa Idli Vada Dosa Pongal Upma Vada Vada Vada"
words = word_tokenize(text)

# Create frequency distribution
freq_dist = nltk.FreqDist(words)
df = pd.DataFrame(freq_dist.items(), columns=["Word", "Frequency"])
print(df)

# WordCloud
wc = WordCloud(width=500, height=300).generate_from_frequencies(freq_dist)
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```

📘 7. spaCy

Theory:

spaCy is a fast and production-ready NLP library. It handles tokenization, POS tagging, Named Entity Recognition, etc.

python

```python
!pip install -q spacy
import spacy

nlp = spacy.load("en_core_web_sm")

doc = nlp("Dosa is popular. Chennai Super Kings won the match.")
for token in doc:
    print(token.text, "|", token.pos_, "|", token.dep_)
```

📘 8. Applications of NLP

Examples of Applications:

1. **Chatbots** — like customer service bots that understand questions and respond
2. **Text Classification** — e.g., filtering South Indian food reviews as positive or negative

✅ Example 1: Simple Sentiment Classifier using TextBlob

```python
review = TextBlob("Idli and Vada are so tasty!")
print("Sentiment:", "Positive" if review.sentiment.polarity > 0 else "Negative")
```

✅ Example 2: Categorize IPL comments

```python
comments = [
    "Mumbai Indians played poorly.",
    "Chennai Super Kings dominated the match!",
]

for c in comments:
    sentiment = TextBlob(c).sentiment.polarity
    label = "Positive" if sentiment > 0 else "Negative"
    print(f"Comment: {c}\nSentiment: {label}\n")
```
