

Natural Language Processing (NLP)

Natural Language Processing (NLP) is a branch of artificial intelligence that focuses on **enabling computers to understand, interpret, and generate human language**. NLP combines linguistics, computer science, and machine learning to analyze textual or spoken data for meaningful insights.

Key Concepts in NLP

1. Tokenization

- Splitting text into smaller units such as words, sentences, or subwords.
- Example: “I love AI” → [“I”, “love”, “AI”]

2. Stop Words Removal

- Removing common words that do not add significant meaning (e.g., “the”, “is”, “and”).

3. Stemming and Lemmatization

- **Stemming:** Reduces words to their root form (e.g., “running” → “run”).
- **Lemmatization:** Reduces words to their dictionary form considering context.

4. Vectorization / Embeddings

- Converts text into numerical representations for machine learning models.
- Techniques:
 - Bag-of-Words (BoW)
 - TF-IDF (Term Frequency-Inverse Document Frequency)
 - Word Embeddings (Word2Vec, GloVe)
 - Contextual embeddings (BERT, GPT)

5. Part-of-Speech (POS) Tagging

- Assigns grammatical tags (noun, verb, adjective) to each word in a sentence.

6. Named Entity Recognition (NER)

- Identifies entities like names, locations, dates, and organizations in text.

7. Sentiment Analysis

- Determines the emotional tone of a text: positive, negative, or neutral.

8. Text Classification

- Categorizes text into predefined classes, e.g., spam detection, topic classification.

9. Sequence Models

- Handles sequential data, commonly using Recurrent Neural Networks (RNNs), LSTMs, or Transformers.
-

Popular Python Libraries for NLP

- **NLTK (Natural Language Toolkit):** Tokenization, stemming, POS tagging.
 - **spaCy:** Industrial-strength NLP library with fast processing.
 - **gensim:** Topic modeling and word embeddings.
 - **Transformers (Hugging Face):** Pre-trained models like BERT, GPT for advanced NLP tasks.
 - **scikit-learn:** Text preprocessing and vectorization for classical ML models.
-

Python Example (Text Preprocessing and Vectorization):

```
import nltk  
from nltk.tokenize import word_tokenize  
from sklearn.feature_extraction.text import TfidfVectorizer  
  
# Sample text  
text = ["I love AI and machine learning.", "Natural Language Processing is fascinating."  
  
# Tokenization  
tokens = [word_tokenize(sentence) for sentence in text]  
print("Tokens:", tokens)
```

```
# TF-IDF Vectorization  
  
vectorizer = TfidfVectorizer()  
  
X = vectorizer.fit_transform(text)  
  
print("TF-IDF Feature Names:", vectorizer.get_feature_names_out())  
  
print("TF-IDF Vectors:\n", X.toarray())
```

Applications of NLP

- Chatbots and virtual assistants (e.g., Siri, Alexa).
- Sentiment analysis for social media or product reviews.
- Machine translation (e.g., Google Translate).
- Text summarization and content recommendation.
- Information retrieval and search engines.

NLP enables computers to **bridge the gap between human language and machine understanding**, powering applications in communication, data analysis, and AI-driven insights.