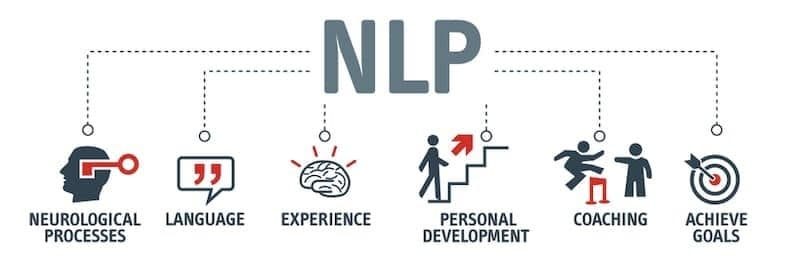
**Natural Language Processing**

Natural Language Processing (NLP) is a subfield of computer science and artificial intelligence that focuses on the interaction between humans and computers using natural language. NLP enables computers to understand, interpret, and generate human language, making it a powerful tool for a wide range of applications, from chatbots and voice assistants to sentiment analysis and text classification.

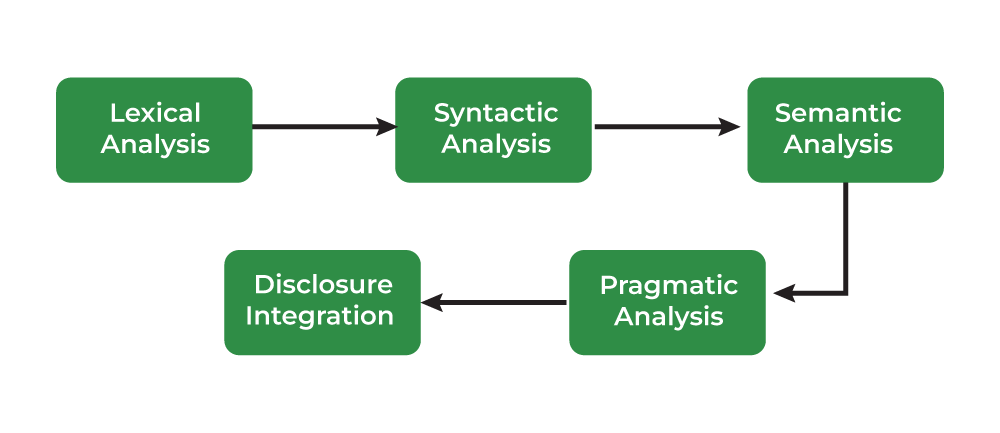
Natural Language Processing (NLP) is the branch of Artificial Intelligence (AI) that gives the ability to machine understand and process human languages. Human languages can be in the form of text or audio format.



**Applications of NLP**

The applications of Natural Language Processing are as follows:

* Voice Assistants like Alexa, Siri, and Google Assistant use NLP for voice recognition and interaction.
* Tools like Grammarly, Microsoft Word, and Google Docs apply NLP for grammar checking and text analysis.
* Information extraction through Search engines such as Google and DuckDuckGo.
* Website bots and customer support chatbots leverage NLP for automated conversations and query handling.
* Google Translate and similar services use NLP for real-time translation between languages.
* Text summarization
* **Phases of Natural Language Processing**



**Libraries for Natural Language Processing**

Some of natural language processing libraries include:

* NLTK (Natural Language Toolkit)
* spaCy
* Transformers (by Hugging Face)

 Before you can analyze that data programmatically, you first need to preprocess it.

The latest version of NLTK is 3.9 so we are going to install it

python -m pip install nltk==3.9

Tokenization

**Tokenization** in**natural language processing (NLP)** is a technique that involves dividing a sentence or phrase into smaller units known as tokens. These tokens can encompass words, dates, punctuation marks, or even fragments of words.

Tokenization is the process of breaking a text into individual words or tokens. NLTK provides a word\_tokenize function that performs this task.

**Filtering Stop Words**

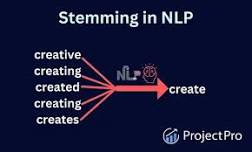
**Stop words** are words that you want to ignore, so you filter them out of your text when you’re processing it. Very common words like 'in', 'is', and 'an' are often used as stop words since they don’t add a lot of meaning to a text in and of themselves.

Stopwords are common words that do not add much meaning to a sentence, such as “the,” “is,” and “and.” NLTK provides a stopwords module that contains a list of stop words for various languages.

**Stemming**

**Stemming** is a text processing task in which you reduce words to their [root](https://simple.wikipedia.org/wiki/Root_(linguistics)), which is the core part of a word. For example, the words “helping” and “helper” share the root “help.” Stemming allows you to zero in on the basic meaning of a word rather than all the details of how it’s being used. NLTK has [more than one stemmer](http://www.nltk.org/howto/stem.html), but you’ll be using the [Porter stemmer](https://www.nltk.org/_modules/nltk/stem/porter.html).

The process of removing affixes from a word so that we are left with the stem of that word is called stemming. For example, consider the words 'run', 'running', and 'runs', all convert into the root word 'run' after stemming is implemented on them.



| **Original word** | **Stemmed version** |
| --- | --- |
| 'Discovery' | 'discoveri' |
| 'discovered' | 'discov' |
| 'discoveries' | 'discoveri' |
| 'Discovering' | 'discov' |

1. **Understemming** happens when two related words should be reduced to the same stem but aren’t.
2. **Overstemming** happens when two unrelated words are reduced to the same stem even though they shouldn’t be.

The [Porter stemming algorithm](https://tartarus.org/martin/PorterStemmer/) dates from 1979, so it’s a little on the older side.

**Tagging Parts of Speech**

**Part of speech** is a grammatical term that deals with the roles words play when you use them together in sentences. Tagging parts of speech, or **POS tagging**, is the task of labeling the words in your text according to their part of speech.

In English, there are eight parts of speech:

| **Part of speech** | **Role** | **Examples** |
| --- | --- | --- |
| Noun | Is a person, place, or thing | mountain, bagel, Poland |
| Pronoun | Replaces a noun | you, she, we |
| Adjective | Gives information about what a noun is like | efficient, windy, colorful |
| Verb | Is an action or a state of being | learn, is, go |
| Adverb | Gives information about a verb, an adjective, or another adverb | efficiently, always, very |
| Preposition | Gives information about how a noun or pronoun is connected to another word | from, about, at |
| Conjunction | Connects two other words or phrases | so, because, and |
| Interjection | Is an exclamation | yay, ow, wow |

Some sources also include the category **articles** (like “a” or “the”) in the list of parts of speech, but other sources consider them to be adjectives.

| **Tags that start with** | **Deal with** |
| --- | --- |
| JJ | Adjectives |
| NN | Nouns |
| RB | Adverbs |
| PRP | Pronouns |
| VB | Verbs |

**Lemmatization**

Lemmatization is the process of reducing words to their base form, or lemma. For example, the word “running” would be reduced to “run.” This can help to group together words with similar meanings. NLTK provides a WordNetLemmatizer class that performs lemmatization.

Like stemming, **lemmatizing** reduces words to their core meaning, but it will give you a complete English word that makes sense on its own instead of just a fragment of a word like 'discoveri'.

**lemma** is a word that represents a whole group of words, and that group of words is called a **lexeme**.

In Natural Language Processing (NLP), chunking is the process of grouping words together to form phrases based on their grammatical roles, like noun phrases (NP) or verb phrases (VP). It's often used to extract information from text, such as named entities (places, people, organizations), and can be done using rule-based systems or machine learning models.

Chunking

What is Chunking?

* Chunking helps identify parts of speech and short phrases within a sentence.
* It groups words that are closely related, often forming noun phrases.
* This process can help extract information from text, such as identifying places, people's names, or other relevant entities.

**Chunk extraction or partial parsing** is a process of meaningful extracting short phrases from the sentence (tagged with Part-of-Speech).   
Chunks are made up of words and the kinds of words are defined using the part-of-speech tags.

**CountVectorizer**is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors (for using in further text analysis).

**What is Sentiment Analysis**

Sentiment analysis is a technique used to determine the emotional tone or sentiment expressed in a text. It involves analyzing the words and phrases used in the text to identify the underlying sentiment, whether it is positive, negative, or neutral.

Sentiment analysis has a wide range of applications, including social media monitoring, customer feedback analysis, and market research.

One of the main challenges in sentiment analysis is the inherent complexity of human language. Text data often contains sarcasm, irony, and other forms of figurative language that can be difficult to interpret using traditional methods.

Chunking

In NLP, chunking breaks down sentences into phrases (chunks) that are more meaningful than individual words. This process helps identify and group phrases, like noun phrases, and extract information from text. In Python, chunking can involve splitting sequences or lists into smaller, manageable portions for processing.

**What is Chunking?**

Chunking is a technique in NLP that groups words together based on their grammatical roles, like part-of-speech tags. A common example is grouping words into noun phrases (NP).

Chinking in Natural Language Processing (NLP) is the process of removing a sequence of tokens from a chunk. It is often used in conjunction with chunking, where you first define what should be included in a chunk and then define what should be excluded.

In Natural Language Processing (NLP), **Named Entity Recognition (NER)** is a technique that identifies and classifies entities in text, such as people, organizations, locations, and dates. It's a subtask of information extraction that transforms unstructured text into structured data, making it valuable for various NLP applications.

What are Named Entities?

Named entities are specific words or phrases that refer to real-world objects, people, or concepts. They are not just any word, but specific and identifiable things like:

* **People:** Barack Obama, Elon Musk
* **Organizations:** Google, Facebook
* **Locations:** New York City, Paris
* **Dates:** May 27, 2025
* **Times:** 9:00 AM, 5 PM
* **Monetary values:** $100, €20
* **Events:** The Super Bowl, the Olympics
* **Products:** iPhone, MacBook

Why is NER important?

* **Information Extraction:**

NER helps extract key information from text, making it easier to organize and analyze data.

n Named Entity Recognition (NER), GPE stands for Geo-Political Entity. It identifies entities that represent locations with political or geographical significance, such as countries, cities, states, or regions with defined boundaries or governance.

**Getting Text to Analyze**

NLTK provides several corpora covering everything from novels hosted by [Project Gutenberg](https://www.gutenberg.org/) to inaugural speeches by presidents of the United States.In order to analyze texts in NLTK, you first need to import them. This requires nltk.download("book").

Concordance

In the context of NLP, a concordance is a collection of word locations along with their context. You can use concordances to find: How many times a word appears.