Natural Language Processing

Zeham Management Technologies BootCamp
by SDAIA

September 2nd, 2024

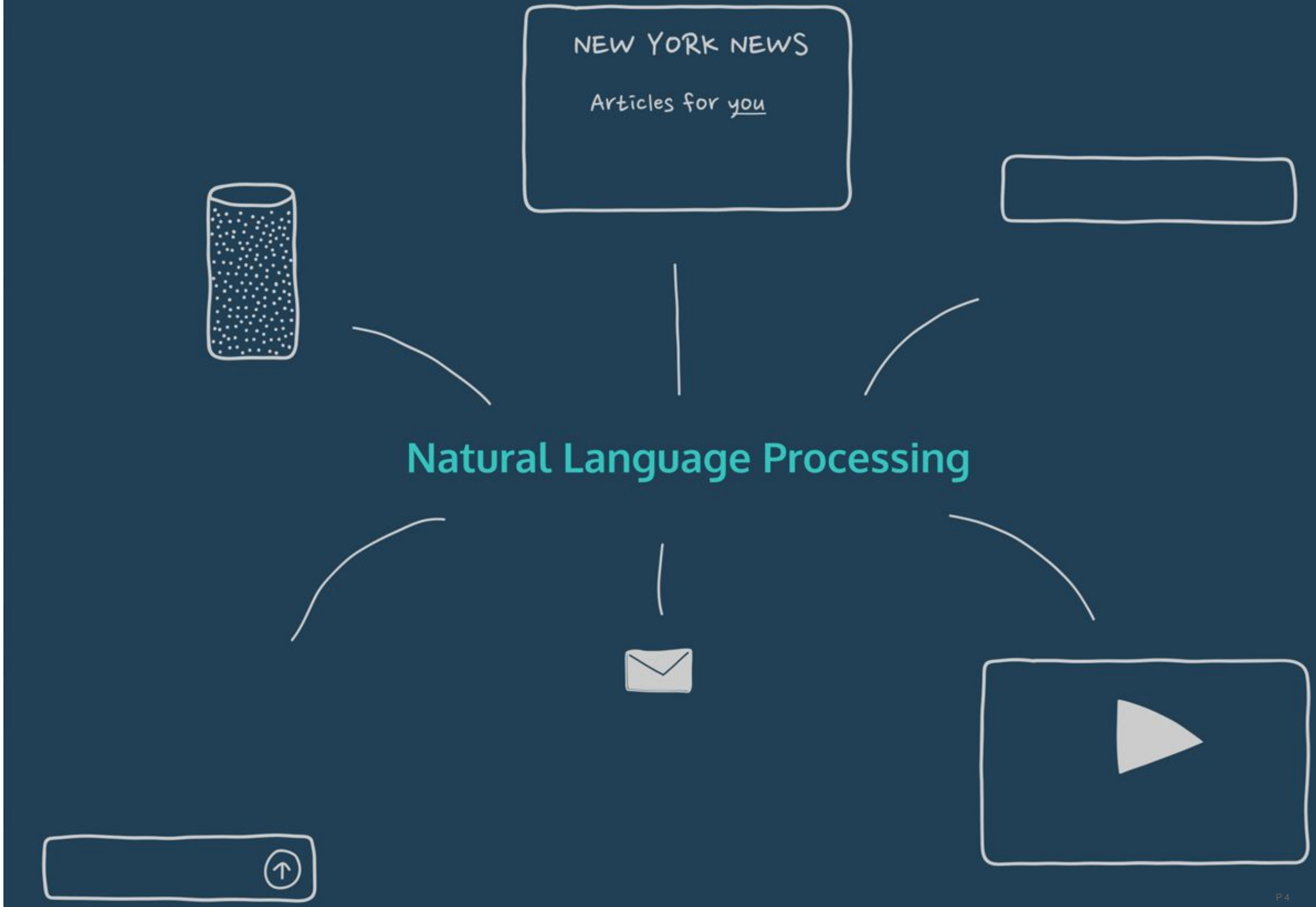
SDAIA
الهيئة السعودية للبيانات
والذكاء الاصطناعي
Saudi Data & AI Authority

Agenda

- Natural language processing (NLP).
- Natural language understanding (NLU).
- Natural language generation (NLG).

Natural language processing (NLP)

We live in a world of NLP



Natural Language Processing

What is NLP anyways?

Natural Language Processing (NLP) is defined as the branch of Artificial Intelligence that provides computers with the capability of understanding text and spoken words, in the same way a human being can.

It incorporates machine learning models, statistics, and deep learning models into computational linguistics i.e. rule-based modeling of human language to allow computers to understand text, spoken words and understands human language, intent, and sentiment.

Why NLP is difficult?

- Understanding human language is considered a difficult task due to its complexity. Also, There are more than 7,100 languages in the world today!
- Spoken language is often riddled with slang, irony, homonyms, and different dialects/inflections, making it For natural language processing to reach its potential, it needs to continuously process more and more
- data—something that can be costly and time-consuming for all but the largest organizations.

Key Fundamentals of Natural Language in NLP

- Syntax

  refers to the rules and structures that govern how words are arranged to form sentences. It helps determine the correct order of words for clear communication.

- Semantics

  deals with the meanings of words, phrases, and sentences. It focuses on understanding the concepts and ideas expressed in language.

- Morphology

  is the study of the structure of words and how they are formed from smaller units called morphemes. For example, the word "unhappiness" consists of three morphemes: "un-", "happy," and "-ness."

Key Fundamentals of Natural Language in NLP

- Pragmatics

  examines how context influences the interpretation of language. It includes aspects like tone, intent, and the situation in which communication occurs, which can affect the meaning of what's being said.

- Phonology and Phonemes

  is the study of sounds in a language. Phonemes are the smallest units of sound that distinguish one word from another. For example, the difference in sound between "bat" and "pat" comes from the phonemes /b/ and /p/.

Key Fundamentals of Natural Language in NLP

- Ambiguity

  occurs when a word or sentence has multiple possible interpretations. For example, the sentence "I saw her duck" could mean either that the speaker observed a person avoiding something or that the speaker saw a woman's pet bird.

- Polysemy

  refers to a single word having multiple related meanings. For instance, the word "bank" can refer to the side of a river or a financial institution, depending on the context.

NLP Ambiguities

There are different types of ambiguities present in natural language:

1. Lexical Ambiguity: It is defined as the ambiguity associated with the meaning of a single word. A single word can have different meanings. Also, a single word can be a noun, adjective, or verb. For example, The word “bank” can have different meanings. It can be a financial bank or a riverbank. Similarly, the word “clean” can be a noun, adverb, adjective, or verb.

NLP Ambiguities

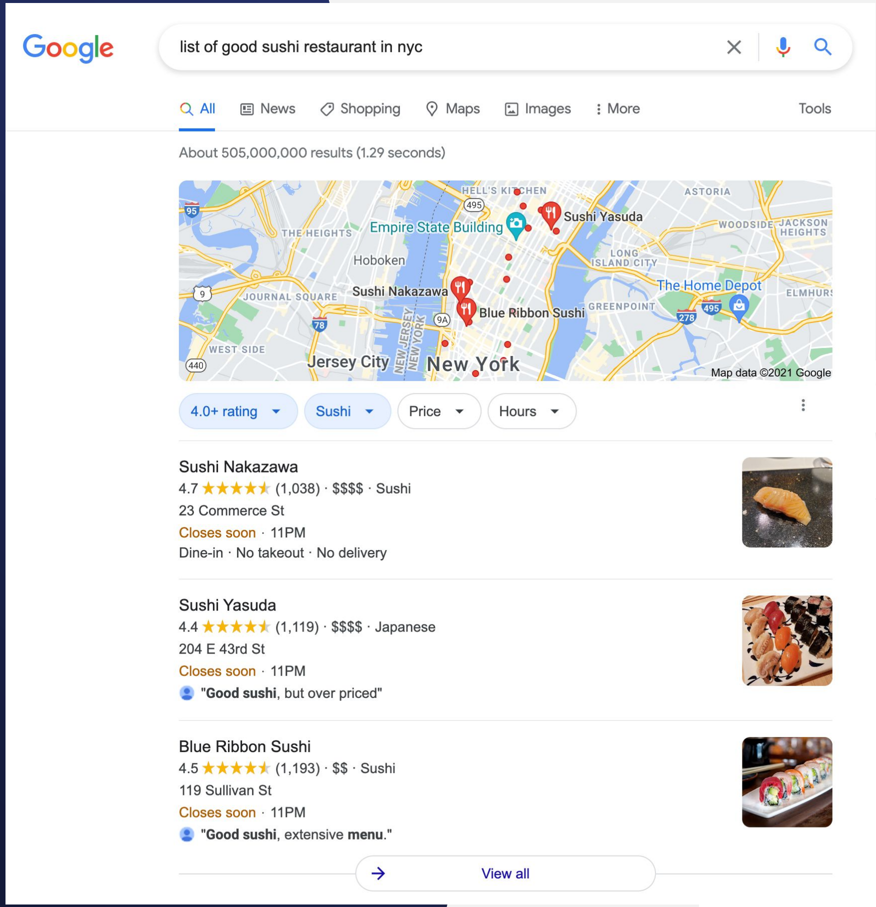
2. Syntactic Ambiguity: It is defined as the ambiguity associated with the way the words are parsed. For example, The sentence “Visiting relatives can be boring.” This sentence can have two different meanings. One is that visiting a relative’s house can be boring. The second is that visiting relatives at your place can be boring.

NLP Ambiguities

3. Semantic Ambiguity: It is defined as ambiguity when the meaning of the words themselves can be ambiguous. For example, The sentence “Mary knows a little french.” In this sentence the word “little french” is ambiguous. As we don’t know whether it is about the language french or a person.

1- Applications

- Information retrieval
- Information extraction
- Question answering



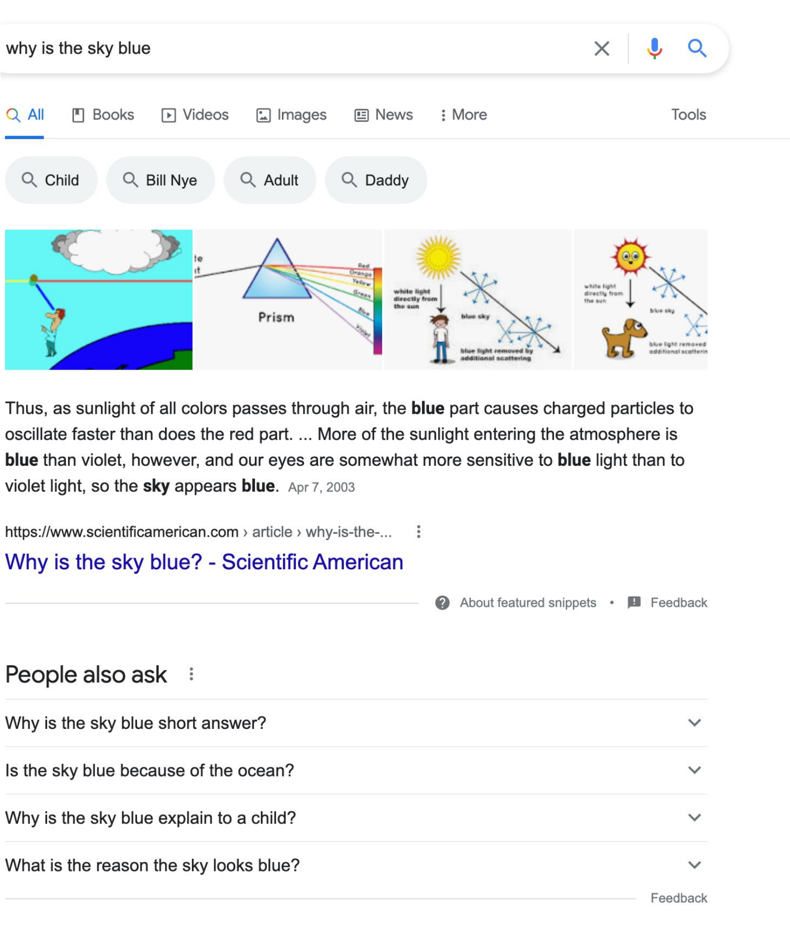
list of good sushi restaurant in nyc

About 505,000,000 results (1.29 seconds)

Sushi Nakazawa
4.7 ★★★★★ (1,038) · $$$$ · Sushi
23 Commerce St
Closes soon · 11PM
Dine-in · No takeout · No delivery

Sushi Yasuda
4.4 ★★★★★ (1,119) · $$$$ · Japanese
204 E 43rd St
Closes soon · 11PM
"**Good sushi**, but over priced"

Blue Ribbon Sushi
4.5 ★★★★★ (1,193) · $$ · Sushi
119 Sullivan St
Closes soon · 11PM
"**Good sushi**, extensive **menu**."



why is the sky blue

Thus, as sunlight of all colors passes through air, the **blue** part causes charged particles to oscillate faster than does the red part. ... More of the sunlight entering the atmosphere is **blue** than violet, however, and our eyes are somewhat more sensitive to **blue** light than to violet light, so the **sky** appears **blue**. Apr 7, 2003

https://www.scientificamerican.com › article › why-is-the-...

Why is the sky blue? - Scientific American

People also ask

Why is the sky blue short answer?

Is the sky blue because of the ocean?

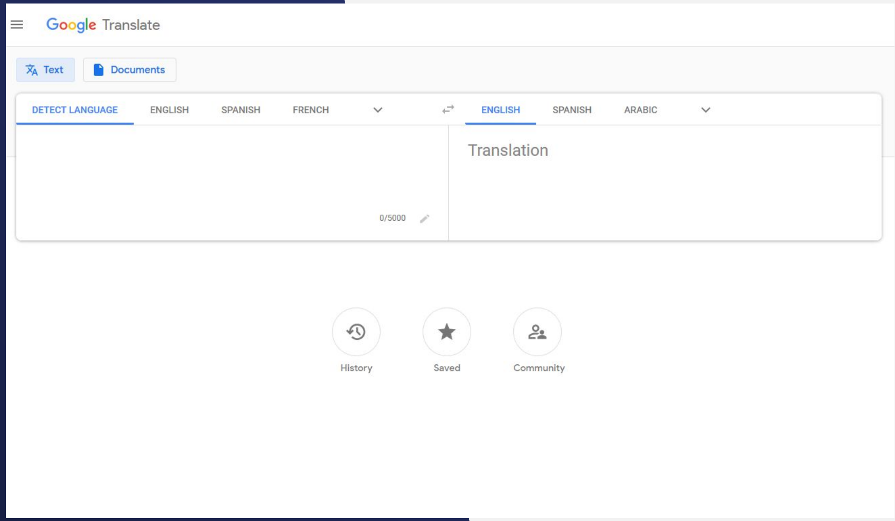
Why is the sky blue explain to a child?

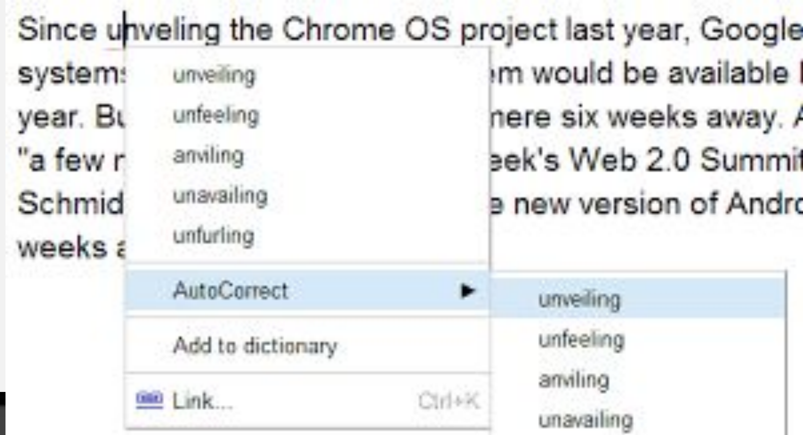
What is the reason the sky looks blue?

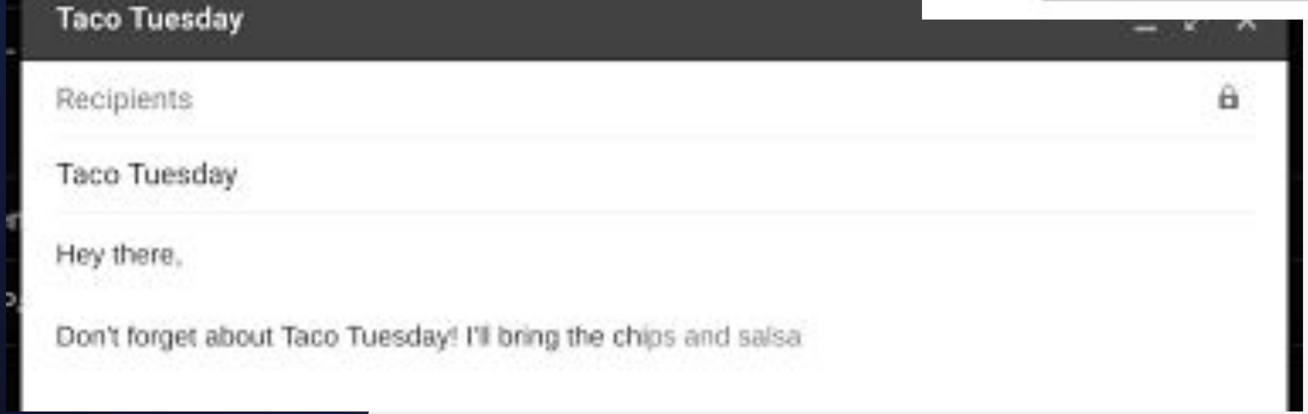
2- Applications

- Machine Translation
- Summarization
- Auto Completion
- Spell Correction

Many More…




Since uhveling the Chrome OS project last year, Google
systems … m would be available
year. Bu … here six weeks away. A
"a few r … eek's Web 2.0 Summit
Schmid … e new version of Andro
weeks a



Taco Tuesday

Hey there,

Don't forget about Taco Tuesday! I'll bring the chips and salsa

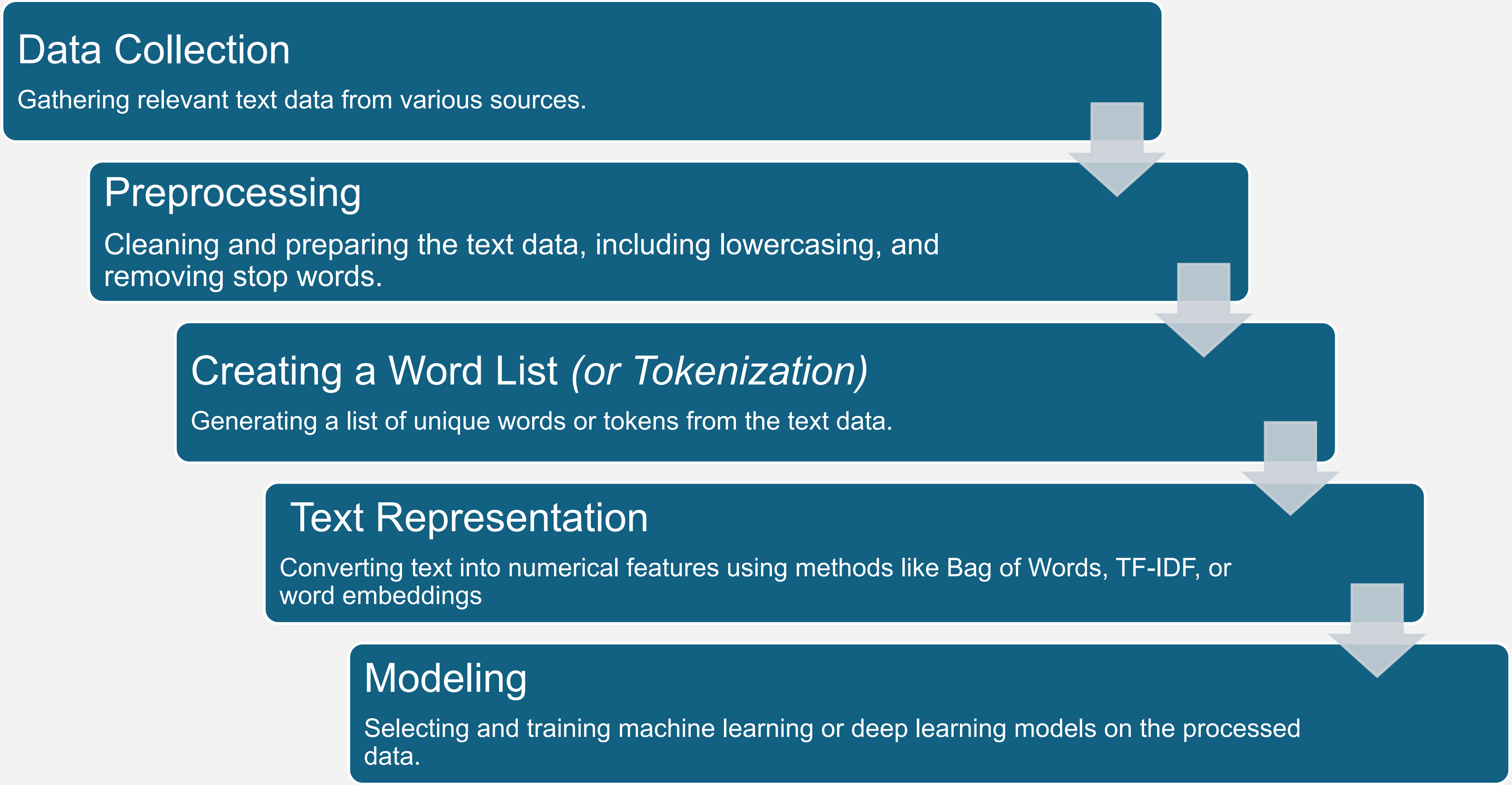
Supervised and Unsupervised NLP

Supervised NLP	Unsupervised NLP
• Supervised NLP methods involve training a model using a dataset with labeled or known input and output pairs. • It means that each piece of input data is associated with the correct output label, which the model learns to predict. • For example, companies might use supervised NLP techniques to train models that categorize documents into specific categories or labels, such as "spam" or "not spam."	• Unsupervised NLP, on the other hand, does not rely on labeled data. Instead, it uses language models to identify patterns and relationships within the text data without predefined labels. • An example of unsupervised NLP is the autocomplete feature, which suggests relevant words or phrases as you type based on patterns it has learned from vast amounts of text data. • This method helps in generating suggestions that make sense in the given context, even without explicit training on labeled examples.

NLP Libraries & Frameworks

Natural Language Toolkit (NLTK) NLTK	Hugging Face Hugging Face	spaCy spaCy	Gensim GENSIM topic modelling for humans	Scikit-Learn scikit learn
A comprehensive Python library for text processing, including tokenization, parsing, and semantic reasoning.	Offers the Transformers library with pre-trained models like BERT and GPT for various NLP tasks. It also provides tools for fine-tuning and a model hub.	An efficient and user-friendly NLP library for tasks like tokenization, part-of-speech tagging, and named entity recognition.	A Python library for topic modeling and document similarity, known for algorithms like Word2Vec and LDA.	It is often used in NLP for tasks such as text classification and feature extraction.

NLP Modeling Steps



Data Collection

Gathering relevant text data from various sources.

Preprocessing

Cleaning and preparing the text data, including lowercasing, and removing stop words.

Creating a Word List *(or Tokenization)*

Generating a list of unique words or tokens from the text data.

Text Representation

Converting text into numerical features using methods like Bag of Words, TF-IDF, or word embeddings

Modeling

Selecting and training machine learning or deep learning models on the processed data.

Text Preprocessing Techniques

Text preprocessing encompasses a series of techniques used to clean, transform, and prepare raw textual data for Natural Language Processing (NLP), Machine Learning (ML), or Deep Learning (DL) tasks. The main steps typically involved in text preprocessing are:

Text preprocessing typically involves:

- Lowercasing
- Cleansing(Remove punc,URLs,stop words,special character…etc)
- Stemming & Lemmatization
- Tokenization
- Text Normalization

The order in which these techniques are applied may vary depending on the needs of the project.

Text Preprocessing Techniques: Lowercasing

Lowercasing: Lowercasing involves converting all characters in a text to their lowercase form to ensure uniformity and reduce redundancy. This process helps standardize text data, making it easier to analyze and compare.

Importance:

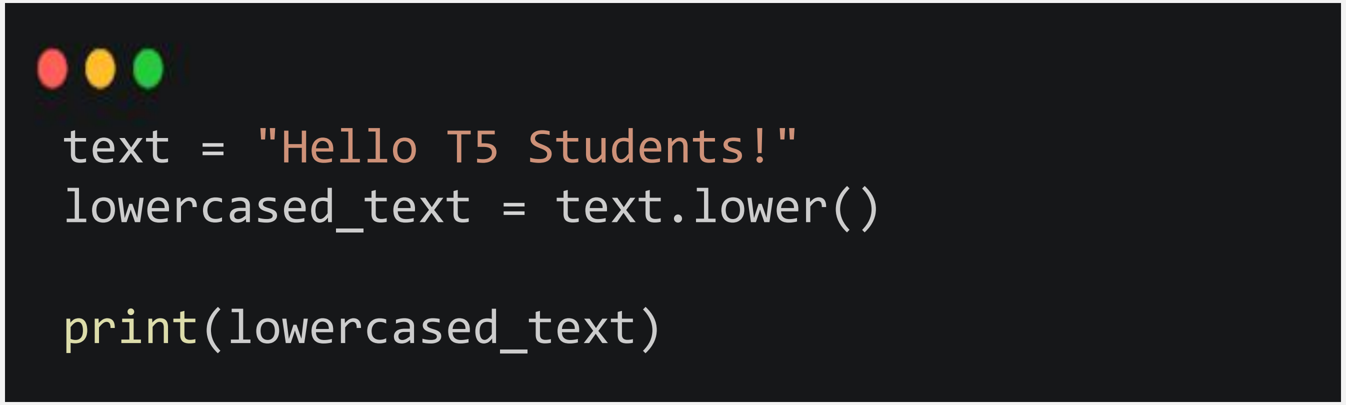
- **Consistency**: Uniform text format reduces variability due to case differences.
- **Redundancy Reduction**: Treats "Apple" and "apple" as the same word, avoiding unnecessary distinctions.
- **Simplifies Analysis**: Facilitates more straightforward text analysis and pattern recognition.

Considerations:

- **Preserve Meaningful Case:** In some contexts, case sensitivity might carry meaning (e.g., "US" vs "us").

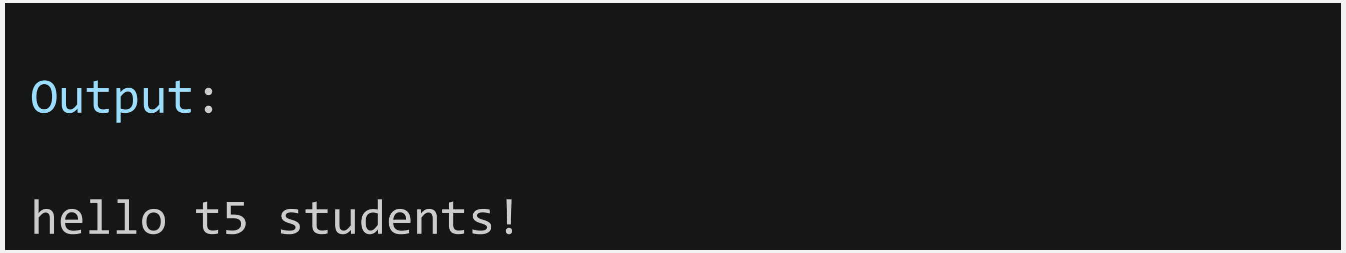
Text Preprocessing Techniques: Lowercasing

This step is implemented so that the algorithm does not treat the same words differently in different situations.



```
text = "Hello T5 Students!"
lowercased_text = text.lower()

print(lowercased_text)
```



```
Output:

hello t5 students!
```

Text Preprocessing Techniques: Removing Punctuation & Spec Characters

Removing punctuation and special characters involves cleaning the text by eliminating symbols that do not contribute to the semantic meaning, thereby simplifying analysis.

Importance:

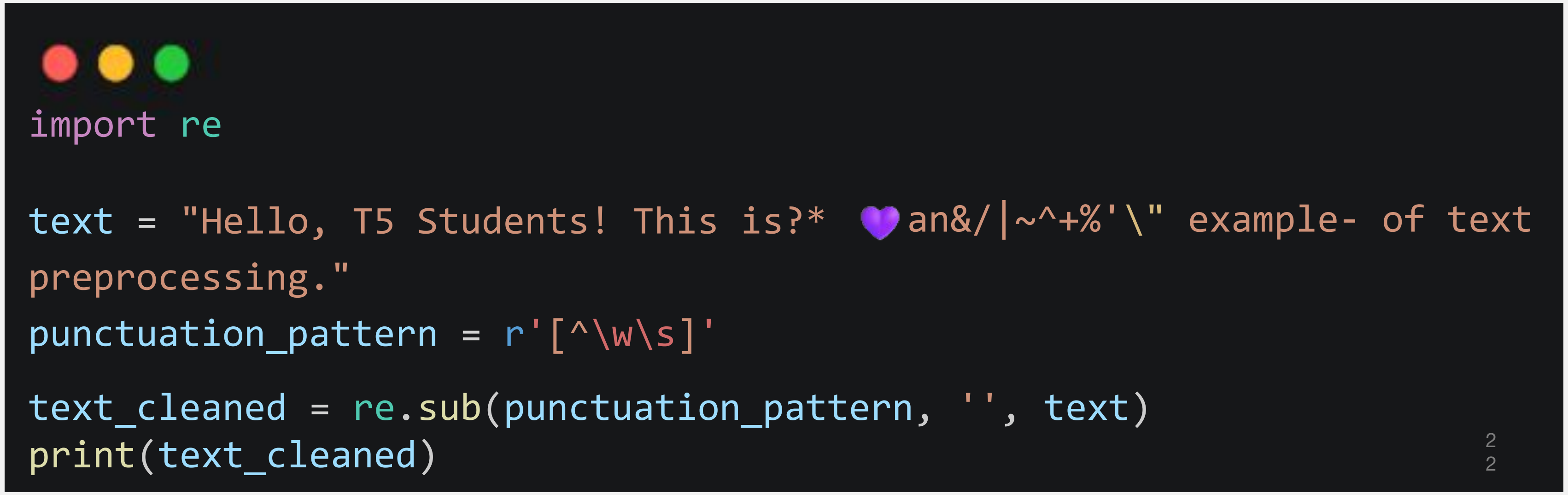
- **Reduces Noise:** Eliminates unnecessary symbols that can clutter the text and complicate processing**.**
- **Improves Accuracy:** Enhances the effectiveness of subsequent text processing steps by focusing on meaningful content.
- **Standardizes Text:** Ensures a more uniform text classification.

Considerations:

- **Context Matters:** Some punctuation may carry meaning in certain contexts (e.g., decimal points, email addresses).

Text Preprocessing Techniques: Removing Punctuation & Spec Characters

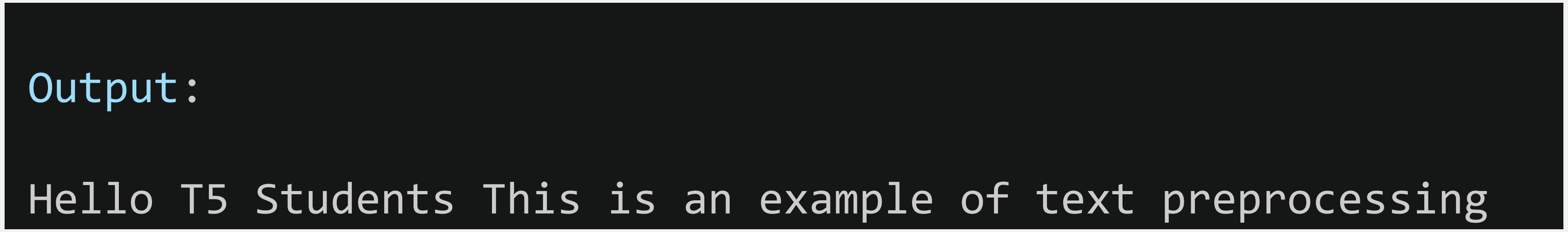
Punctuation removal is a text preprocessing step where you remove all punctuation marks *(such as periods, commas, exclamation marks, emojis etc.)* from the text to simplify it and focus on the words themselves.



```python
import re

text = "Hello, T5 Students! This is?* 💜an&/|~^+%'\" example- of text preprocessing."
punctuation_pattern = r'[^\w\s]'

text_cleaned = re.sub(punctuation_pattern, '', text)
print(text_cleaned)
```



```
Output:

Hello T5 Students This is an example of text preprocessing
```

Text Preprocessing Techniques: Stop-words Removal

Stop-words are common words (e.g., "the," "and," "in") that are often filtered out in text processing because they do not carry significant meaning.

Importance:

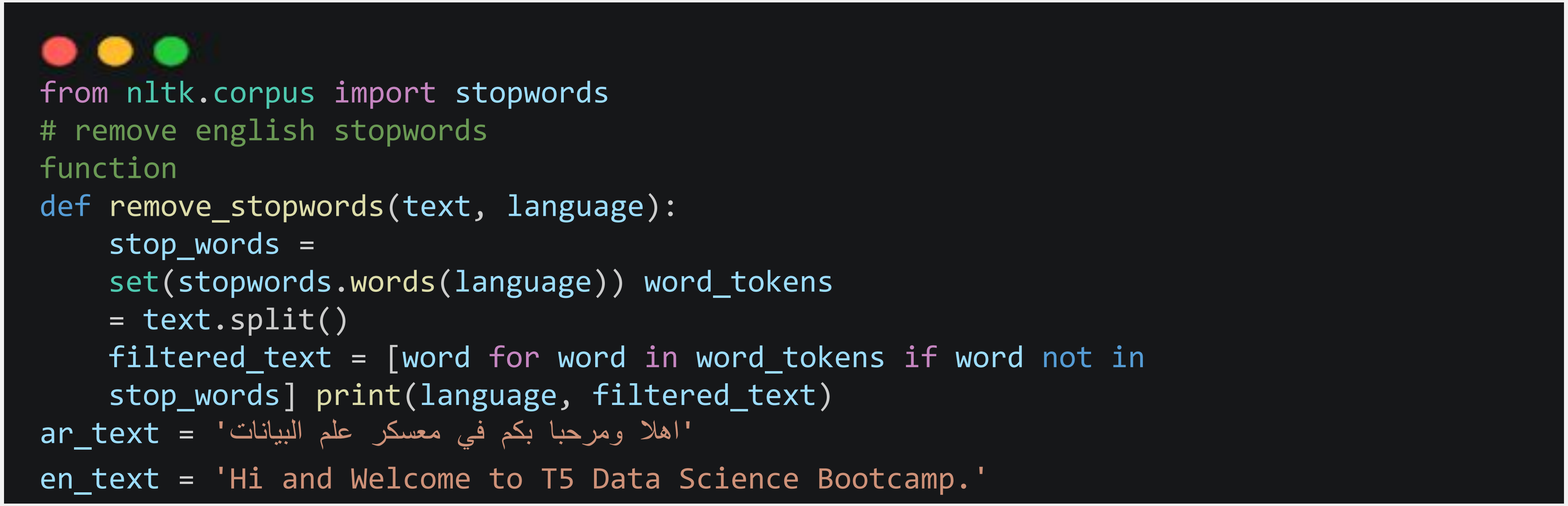
- **Reduces Noise:** By removing common but non-essential words, it minimizes distractions and helps focus on the core content.
- **Improves Efficiency:** Speeds up text processing by reducing the size of the dataset.
- **Enhances Analysis:** Helps in highlighting more meaningful words and phrases.

Considerations:

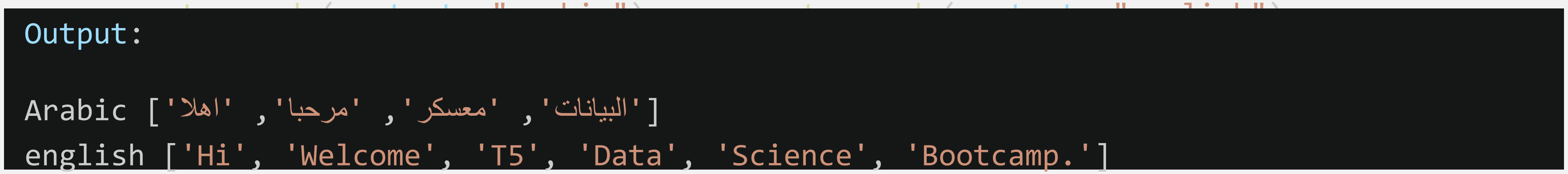
- **Context Sensitivity:** In some contexts, stop-words may be relevant for understanding the text (e.g., sentiment analysis, syntactic structures).

Text Preprocessing Techniques: Stop-words Removal

Stopwords are words that don’t contribute to the meaning of a sentence. The NLTK library has a set of stopwords, and we can use these to remove stopwords from our text and return a list of word allowing the focus on the important words.



```
from nltk.corpus import stopwords
# remove english stopwords
function
def remove_stopwords(text, language):
    stop_words =
    set(stopwords.words(language)) word_tokens
    = text.split()
    filtered_text = [word for word in word_tokens if word not in
    stop_words] print(language, filtered_text)
ar_text = 'اهلا ومرحبا بكم في معسكر علم البيانات'
en_text = 'Hi and Welcome to T5 Data Science Bootcamp.'
```



```
Output:

Arabic ['اهلا', 'مرحبا', 'معسكر', 'البيانات']
english ['Hi', 'Welcome', 'T5', 'Data', 'Science', 'Bootcamp.']
```

Text Preprocessing Techniques: Removal of URLs

URLs are web addresses that can appear in text and may not contribute to the analysis of the content.

Importance:

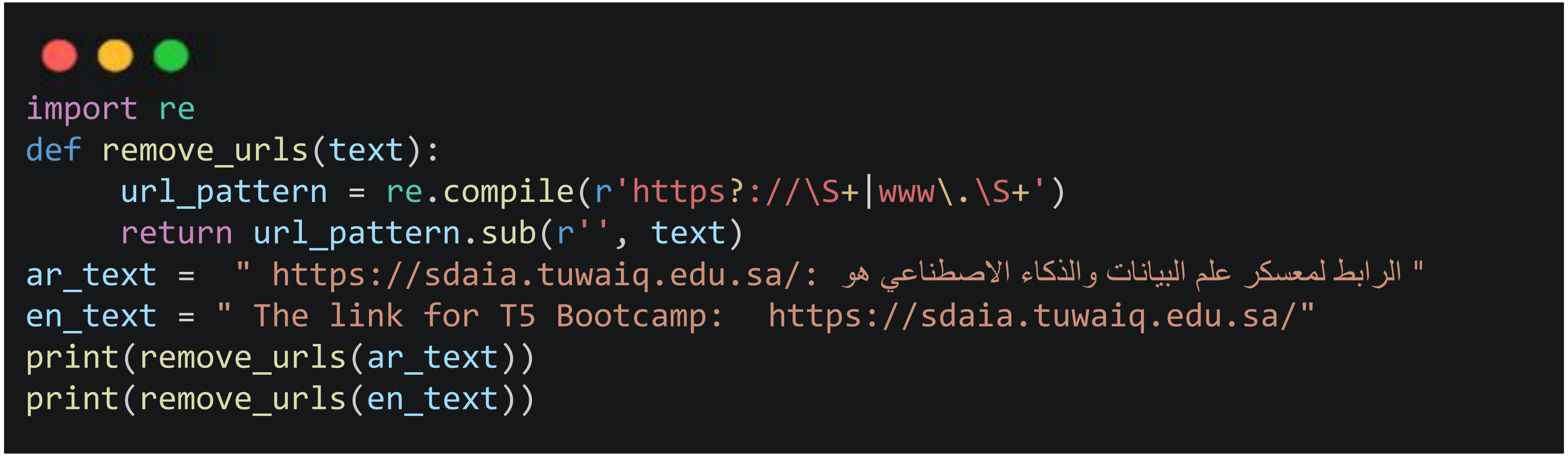
- **Reduces Noise:** Eliminates web links that can clutter the text and distract from the main content.
- **Enhances Processing:** Simplifies text by removing elements that are not relevant to the semantic analysis.
- **Prevents Errors:** Avoids potential issues caused by URLs, such as incorrect tokenization or irrelevant data inclusion.

Considerations:

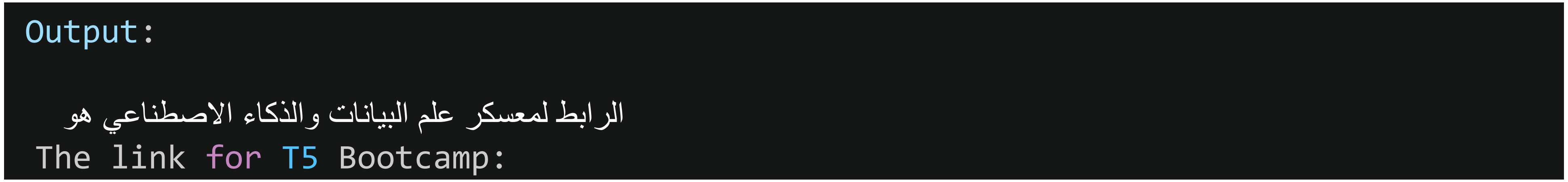
- **Context Awareness:** In some contexts, URLs may contain valuable information (e.g., research papers or references). Ensure that URL removal aligns with the goals of the analysis.
- **Accuracy:** Ensure that URL removal does not inadvertently affect other parts of the text, especially if the URLs are embedded within important content.

Text Preprocessing Techniques: Removal of URLs

This preprocessing step is to remove any URLs present in the data.



```python
import re
def remove_urls(text):
     url_pattern = re.compile(r'https?://\S+|www\.\S+')
     return url_pattern.sub(r'', text)
ar_text =  " https://sdaia.tuwaiq.edu.sa/: الرابط لمعسكر علم البيانات والذكاء الاصطناعي هو "
en_text = " The link for T5 Bootcamp:   https://sdaia.tuwaiq.edu.sa/"
print(remove_urls(ar_text))
print(remove_urls(en_text))
```



```
Output:

  الرابط لمعسكر علم البيانات والذكاء الاصطناعي هو
 The link for T5 Bootcamp:
```

Conclusion

- NLP has revolutionized how we interact with technology, making it possible for machines to understand and generate human language. As the technology evolves, it will continue to play a crucial role in bridging the gap between humans and computers.

Tutorial

7-Natural Language Processing/1-Introduction to Natural Language Processing

(NLP)LAB/NLP.ipynb

Natural language understanding (NLU)

What is Natural Language Understanding?

- Natural language understanding (NLU) is a branch of artificial intelligence (AI) that uses computer software to understand input in the form of sentences using text or speech.
- NLU enables computers to understand the sentiments expressed in a natural language used by humans, without the formalized syntax of computer languages.
- NLU also enables computers to communicate back to humans in their own languages.

Source

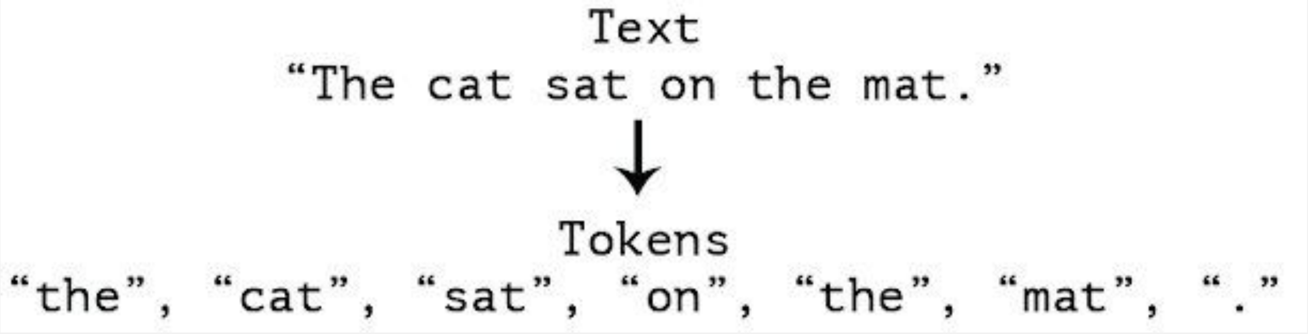
Common NLU tasks

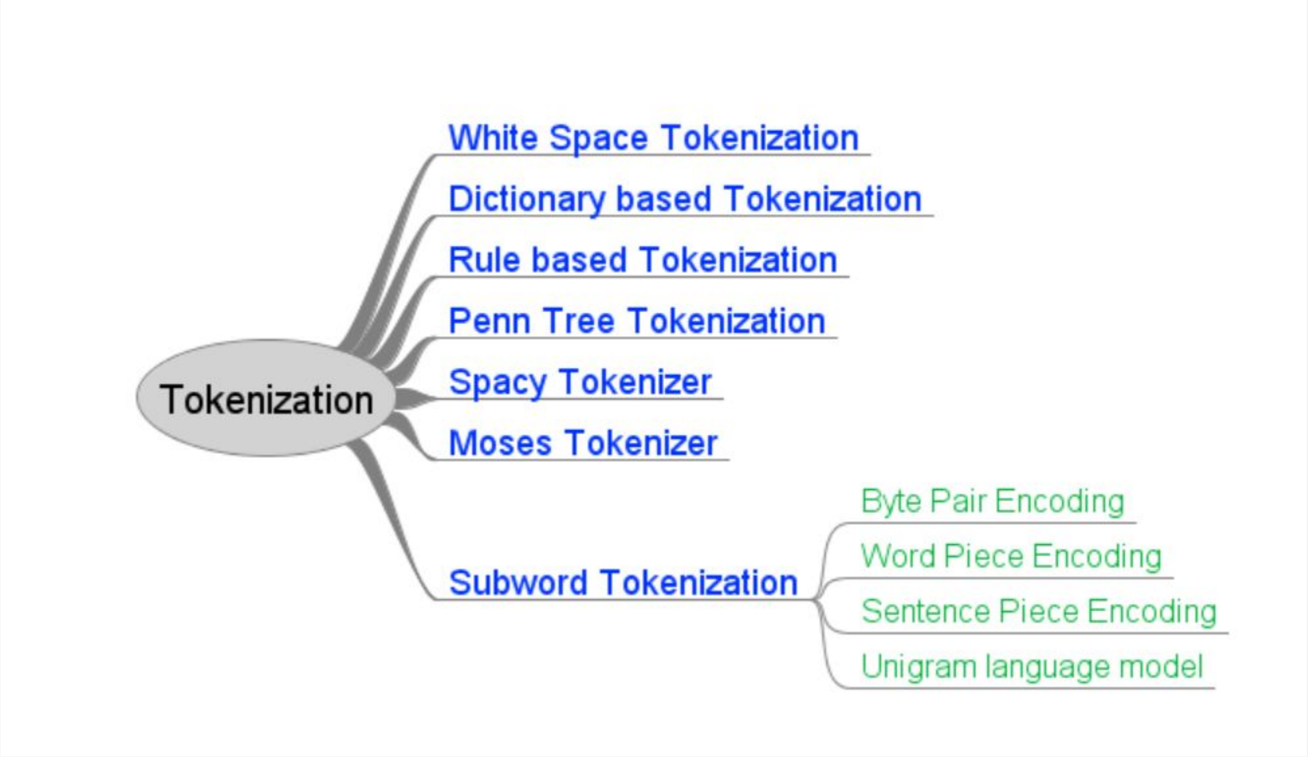
- **Tokenization**
- POS tagging
- Syntactic parsing
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification

- Tokenization is the process of breaking down a text into individual units called tokens.
- Tokens are typically words, but can also be phrases or even individual characters, depending on the application.
- Tokenization is a crucial step in natural language processing tasks such as machine translation, sentiment analysis, and named entity recognition.

Common NLU Tasks

- **Tokenization**
- POS tagging
- Syntactic parsing
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification


Text
“The cat sat on the mat.”
Tokens
“the”, “cat”, “sat”, “on”, “the”, “mat”, “.”



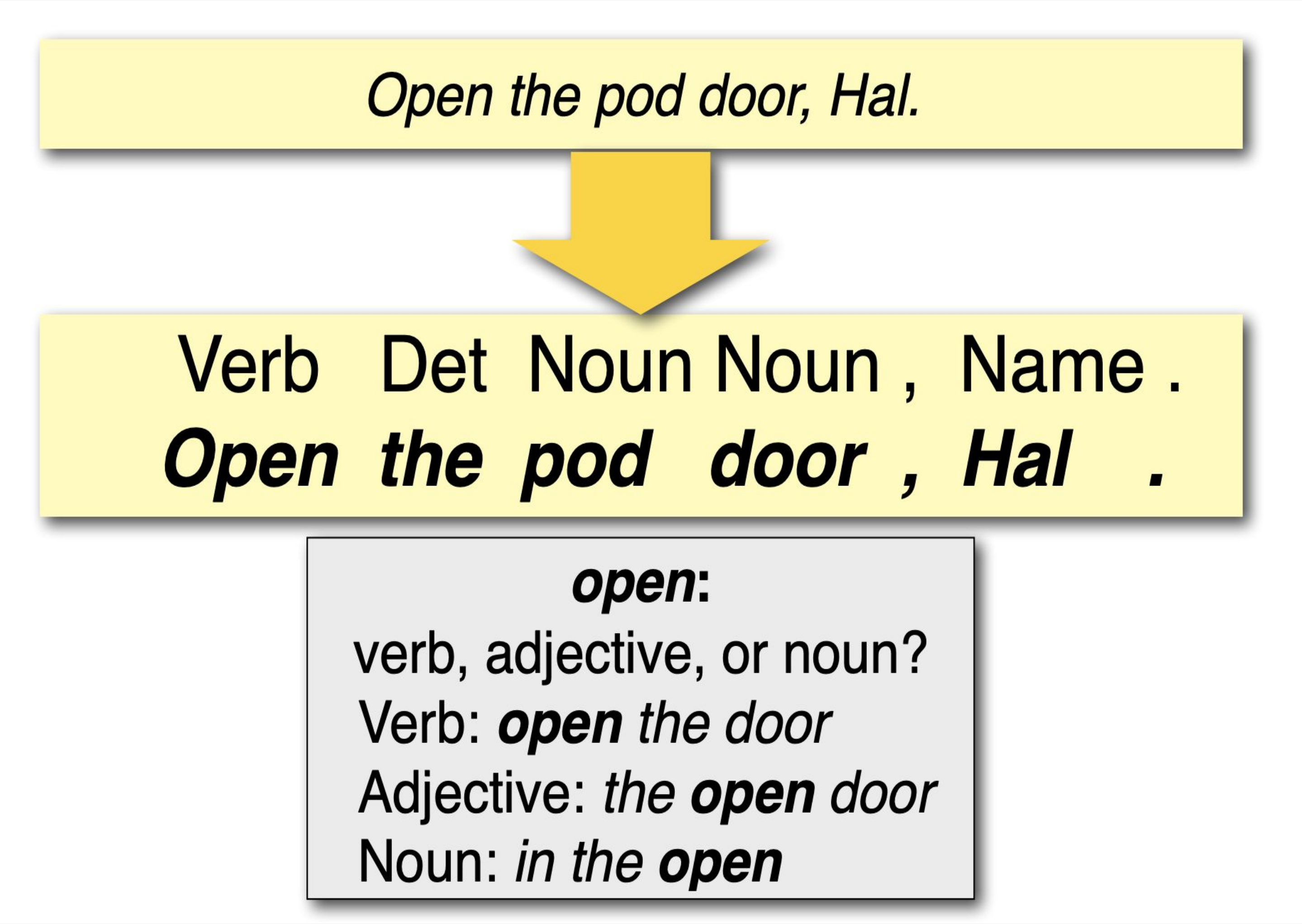
Common NLU Tasks

- Tokenization
- **POS tagging**
- Syntactic parsing
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification

- POS stands for Part-of-Speech, which is a linguistic term used to describe the grammatical category of a word in a sentence.
- POS tagging is the process of assigning each word in a text with its corresponding POS category, such as noun, verb, adjective, or adverb.
- POS tagging is a critical component in various natural language processing tasks, including text-to-speech conversion, information retrieval, and machine translation.

Common NLU Tasks

- Tokenization
- **POS tagging**
- Syntactic parsing
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification



Open the pod door, Hal.

Verb Det Noun Noun , Name .

Open the pod door , Hal .

***open*:**
verb, adjective, or noun?
Verb: ***open** the door*
Adjective: *the **open** door*
Noun: *in the **open***

Common NLU Tasks

- Tokenization
- POS tagging
- **Syntactic parsing**
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification

- Syntactic parsing is the process of analyzing the grammatical structure of a sentence to determine its syntactic components, such as nouns, verbs, adjectives, and adverbs.
- It involves identifying the parts of speech of each word in the sentence and grouping them together into phrases and clauses based on their syntactic relationships.
- Syntactic parsing is used in various natural language processing applications, including text-to-speech conversion, machine translation, and information retrieval.

Common NLU Tasks

- Tokenization
- POS tagging
- **Syntactic parsing**
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification

- POS tagging is the process of labeling individual words in a sentence with their part of speech, such as noun, verb, adjective, or adverb, while syntactic parsing involves analyzing the relationships between the words to determine the overall grammatical structure of the sentence.
- For example, consider the sentence "John eats pizza." POS tagging would label "John" as a proper noun and "eats" as a verb, while syntactic parsing would identify "John" as the subject of the verb "eats" and "pizza" as the object of the verb.
- In short, POS tagging is concerned with the individual words, while syntactic parsing focuses on the overall sentence structure.

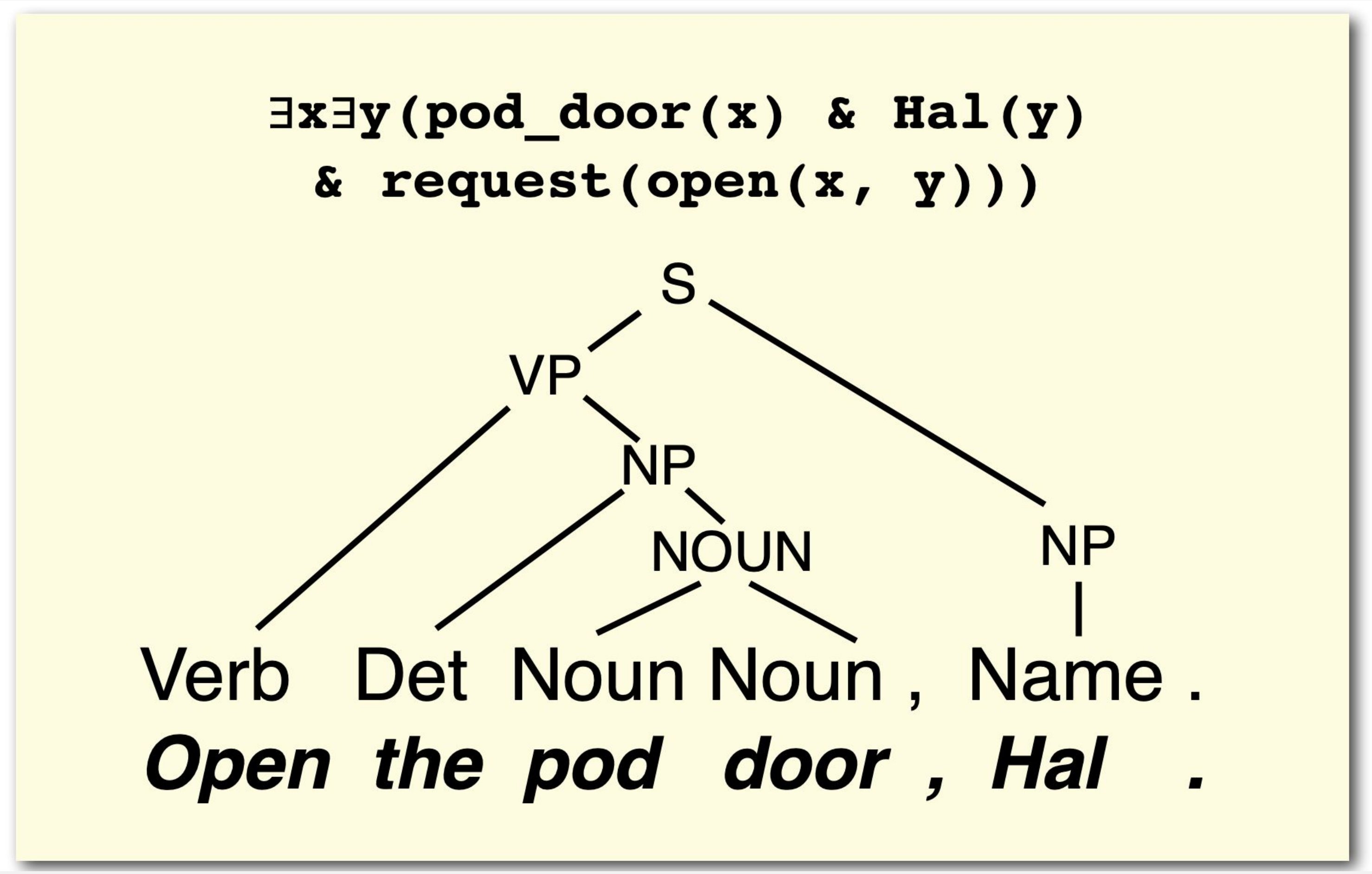
Common NLU Tasks

- Tokenization
- POS tagging
- Syntactic parsing
- **Semantic analysis**
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification

- Semantic analysis is the process of extracting the meaning of a text by analyzing the relationships between words and phrases in a sentence.
- It involves identifying the underlying concepts and ideas conveyed by the text and representing them in a structured form, such as a knowledge graph or ontology.
- Semantic analysis is used in various natural language processing applications, including question answering, information retrieval, and chatbots, to enable more accurate and intelligent responses.

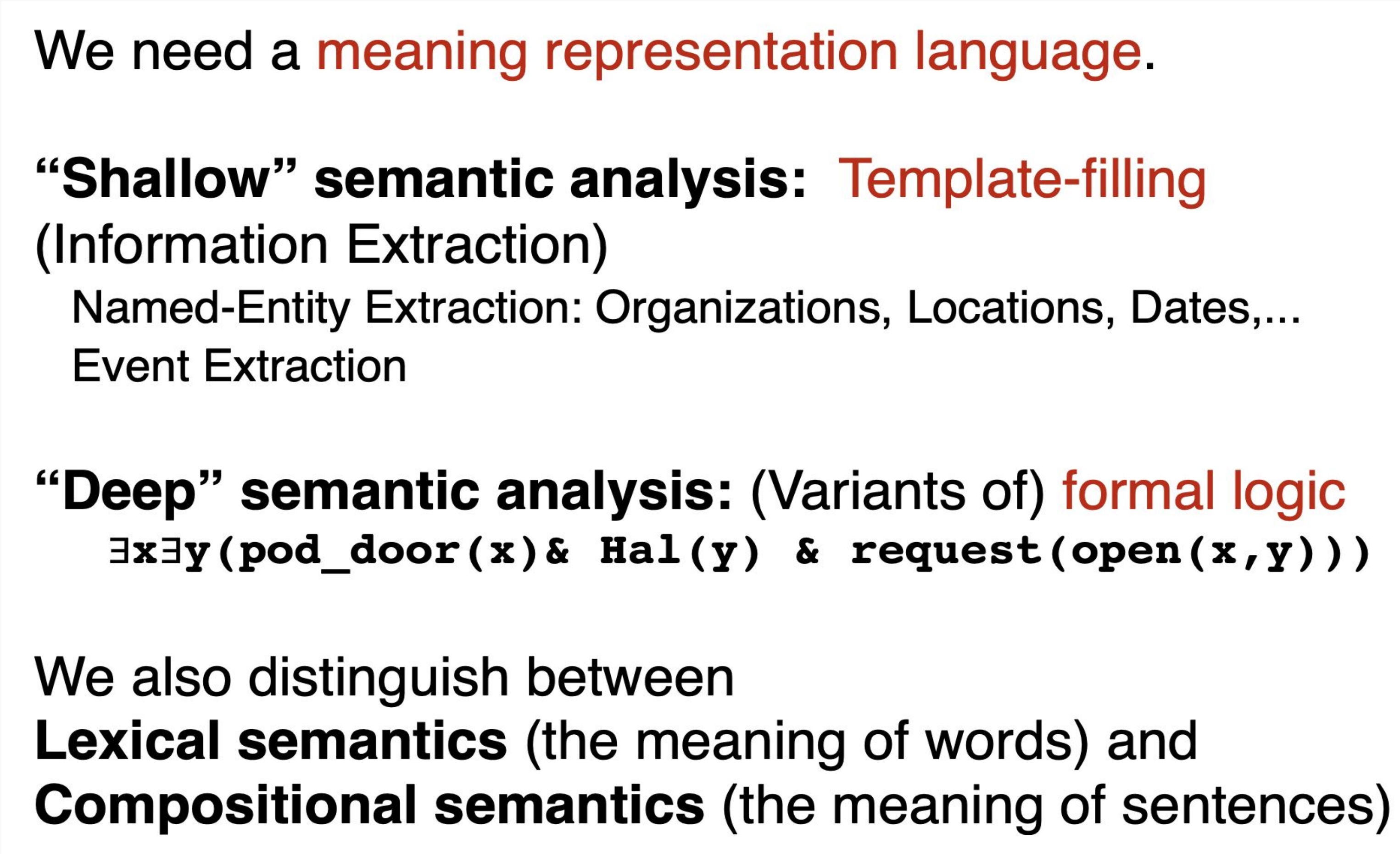
Common NLU Tasks

- Tokenization
- POS tagging
- Syntactic parsing
- **Semantic analysis**
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification


∃x∃y(pod_door(x) & Hal(y)
& request(open(x, y)))
S
VP
NP
NOUN
NP
Verb Det Noun Noun , Name .
Open the pod door , Hal .

Common NLU Tasks

- Tokenization
- POS tagging
- Syntactic parsing
- **Semantic analysis**
- Coreference resolution
- Named Entity Recognition (NER)
- Text classification



We need a meaning representation language.

“Shallow” semantic analysis: Template-filling (Information Extraction)

Named-Entity Extraction: Organizations, Locations, Dates,...

Event Extraction

“Deep” semantic analysis: (Variants of) formal logic

```
∃x∃y(pod_door(x)& Hal(y) & request(open(x,y)))
```

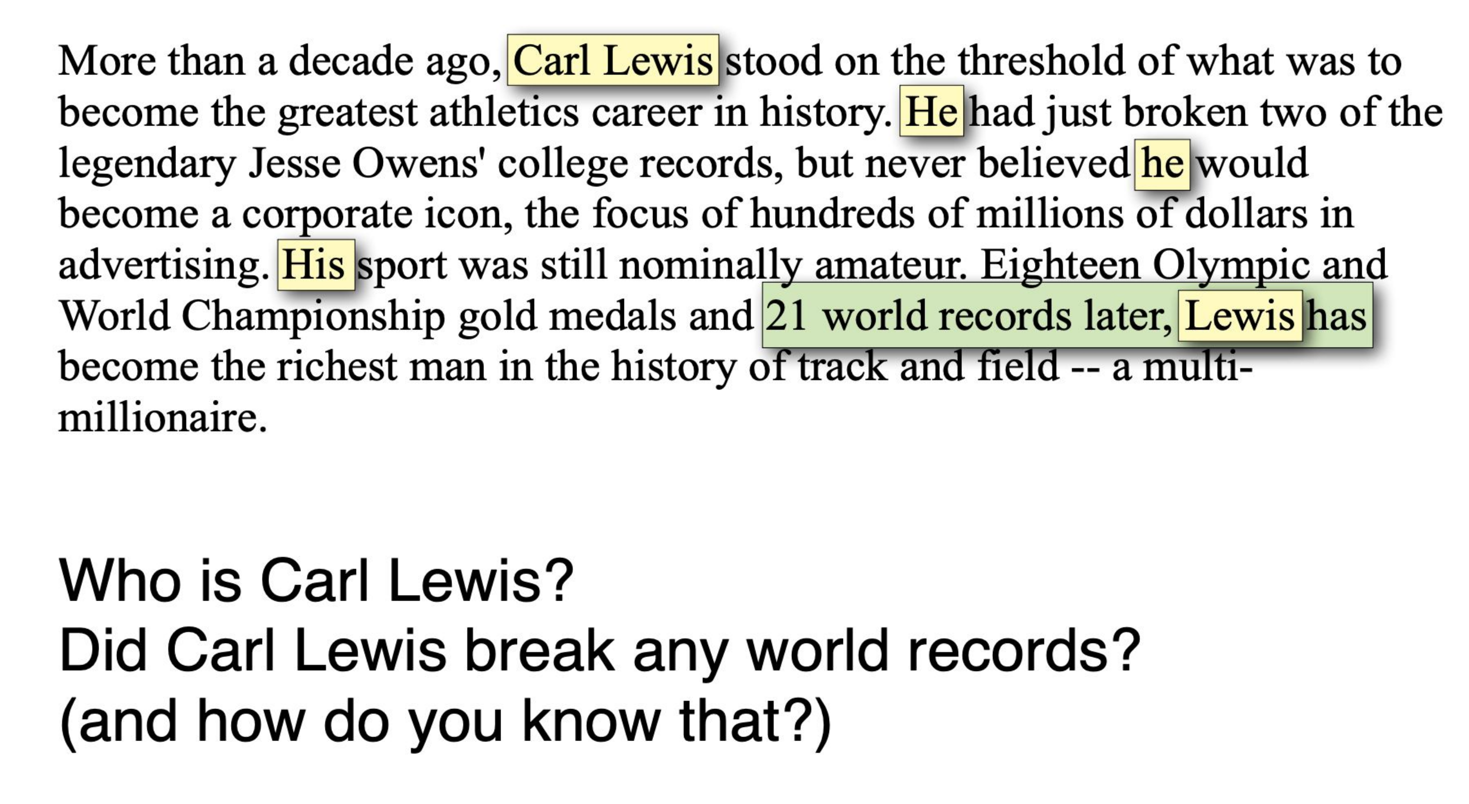
We also distinguish between

Lexical semantics (the meaning of words) and

Compositional semantics (the meaning of sentences)

Common NLU Tasks

- Tokenization
- POS tagging
- Syntactic parsing
- Semantic analysis
- **Coreference resolution**
- Named Entity Recognition (NER)
- Text classification



More than a decade ago, Carl Lewis stood on the threshold of what was to become the greatest athletics career in history. He had just broken two of the legendary Jesse Owens' college records, but never believed he would become a corporate icon, the focus of hundreds of millions of dollars in advertising. His sport was still nominally amateur. Eighteen Olympic and World Championship gold medals and 21 world records later, Lewis has become the richest man in the history of track and field -- a multi-millionaire.

Who is Carl Lewis?
Did Carl Lewis break any world records?
(and how do you know that?)

Common NLU Tasks

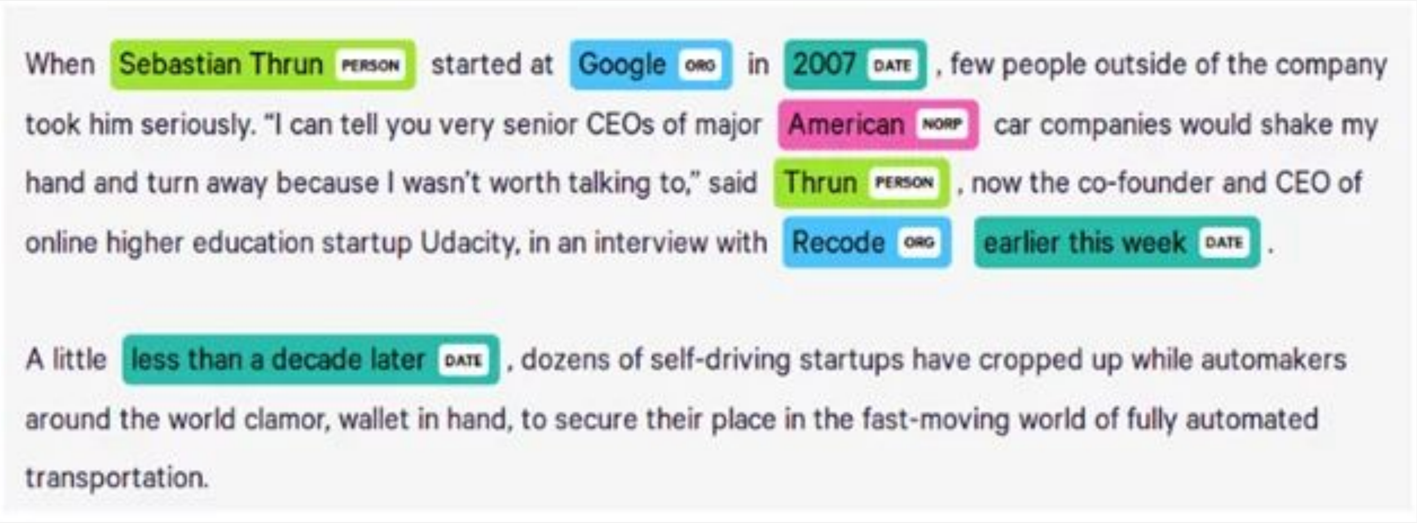
- Tokenization
- POS tagging
- Syntactic parsing
- Semantic analysis
- **Coreference resolution**
- Named Entity Recognition (NER)
- Text classification

- Coreference resolution is the task of identifying all the expressions (e.g., pronouns, names) in a text that refer to the same entity, and linking them together.
- It is a crucial task in natural language processing as it enables a system to maintain a consistent representation of entities throughout a document, enabling more accurate information extraction and text understanding.

Common NLU Tasks

- Tokenization
- POS tagging
- Syntactic parsing
- Semantic analysis
- Coreference resolution
- **Named Entity Recognition (NER)**
- Text classification

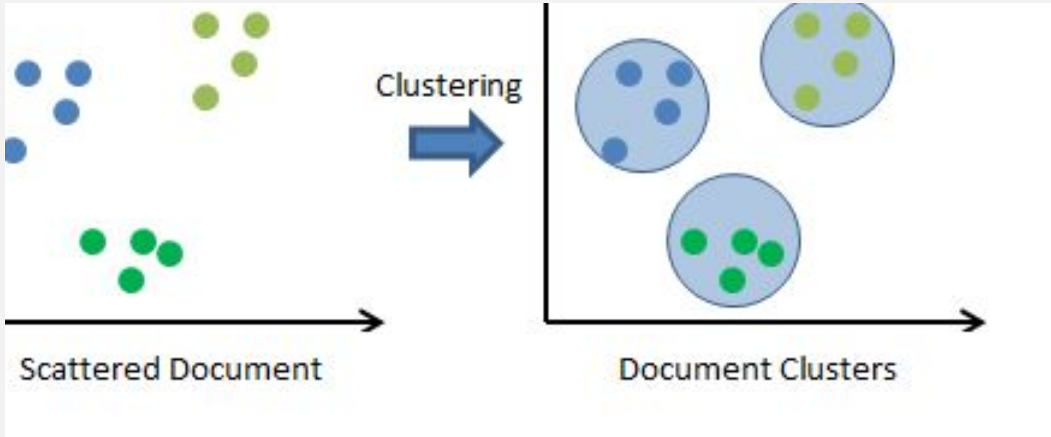
Named entity recognition (NER) is the process of identifying and categorizing named entities in a text, such as people, organizations, locations, and dates.



When Sebastian Thrun PERSON started at Google ORG in 2007 DATE , few people outside of the company took him seriously. "I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I wasn't worth talking to," said Thrun PERSON , now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode ORG earlier this week DATE .

A little less than a decade later DATE , dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

Common NLU Tasks



- Tokenization
- POS tagging
- Syntactic parsing
- Semantic analysis
- Coreference resolution
- Named Entity Recognition (NER)
- **Text classification**

- text classification is the process of converting unstructured text data into a structured format that can be used for natural language processing tasks.
- It involves selecting a suitable representation scheme, such as bag-of-words, word embeddings, or topic models, to capture the key features and characteristics of the text data in a numerical form that can be processed by machine learning algorithms.

Text Preprocessing Techniques: Tokenization

Types of Tokenization:

- **Word Tokenization**

Description: Splits text into individual words, suitable for languages with clear word boundaries like English. Example:

Input: "Natural Language Processing is fascinating!”

Output: ["Natural", "Language", "Processing", "is", "fascinating!"]

- **Character Tokenization**

Description: Segments text into individual characters, beneficial for languages without clear word boundaries or tasks like spelling correction. Example:

Input: "Hello, world!”

Output: ['H', 'e', 'l', 'l', 'o', ',', ' ', 'w', 'o', 'r', 'l', 'd', '!']

Text Preprocessing Techniques: Tokenization

Types of Tokenization:

- **Sub-word Tokenization**

Description: Divides text into units larger than a single character but smaller than a full word, striking a balance between word and character tokenization. Useful for languages forming meaning by combining smaller units or handling out-of-vocabulary words in NLP tasks. Example:

Input: "Chatbots are becoming increasingly popular.”

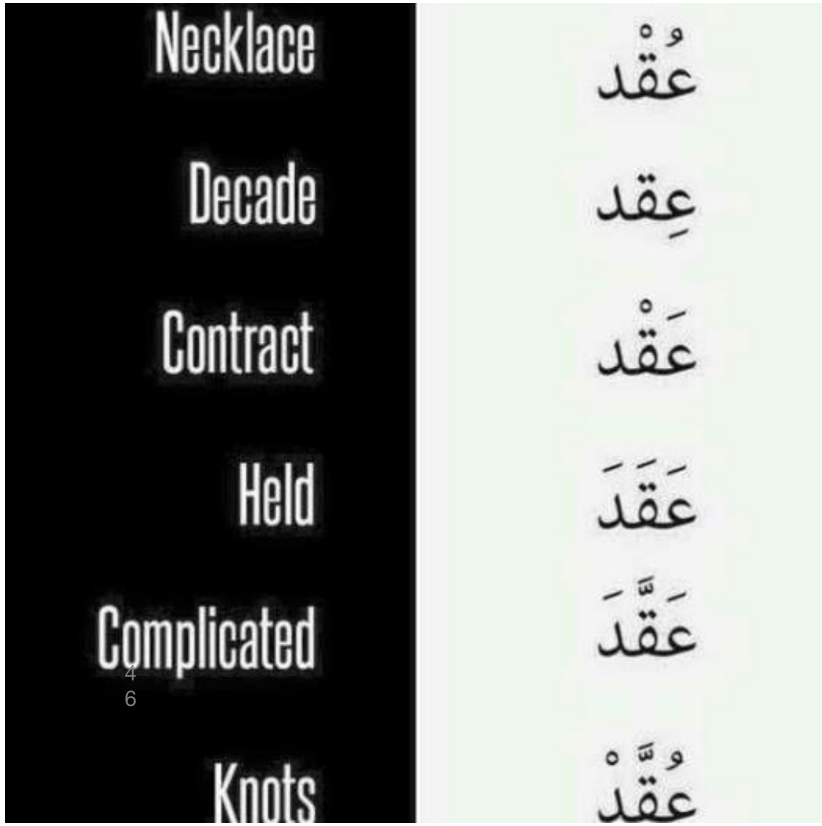
Output: ["Chat", "bots", "are", "becoming", "increasingly", "popular", "."]

Text Preprocessing Techniques: Tokenization

Challenges and Limitations of the tokenization task:

- In general, tokenization is commonly applied to text corpora written in languages like English or French, where words are separated by spaces or punctuation marks that define sentence boundaries.

- However, tokenization of Arabic text presents unique challenges due to the language's complex morphology. Unlike English or French, Arabic features rich inflectional and derivational processes, which can make the separation of words and sentences more intricate.

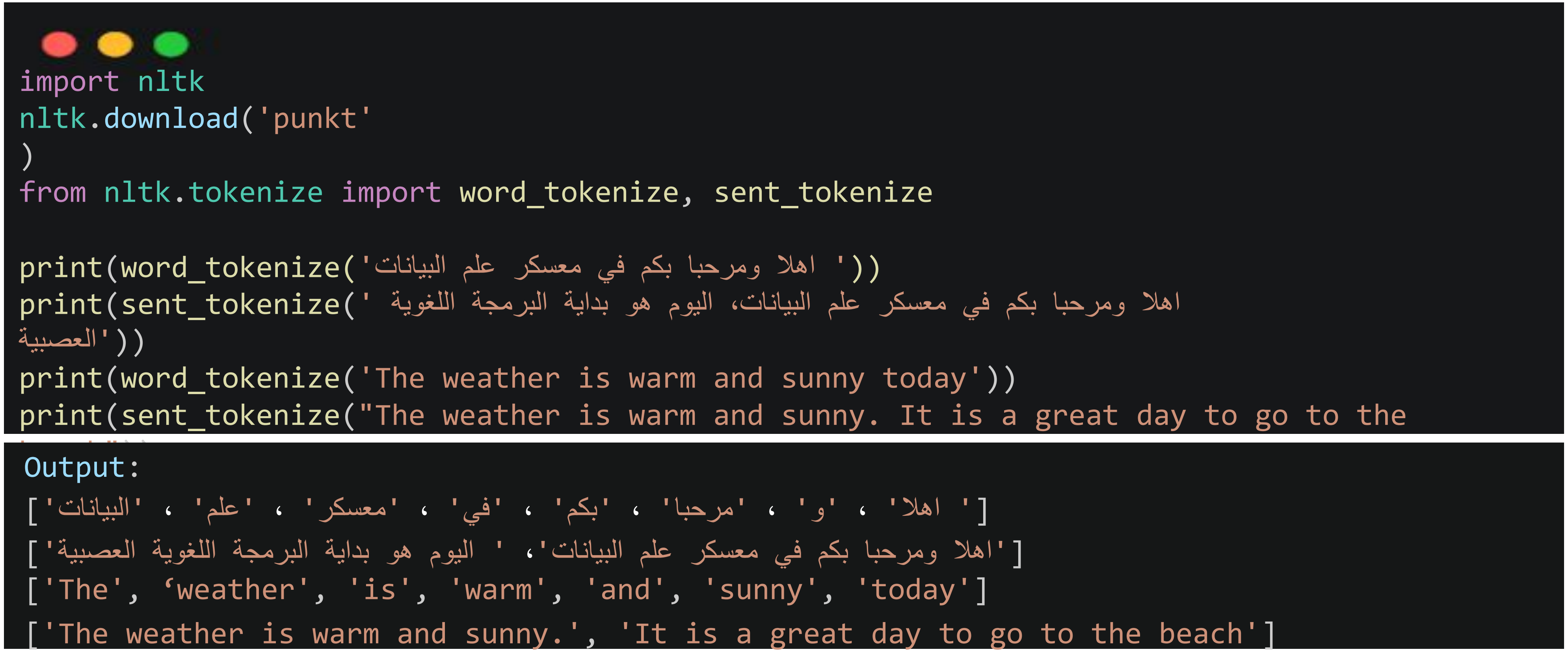


Necklace	عُقْد
Decade	عِقد
Contract	عَقْد
Held	عَقَدَ
Complicated	عَقَّدَ
Knots	عُقَّدْ

For example, a single Arabic word may contain up to six different tokens like the word (“عقد”)

Text Preprocessing Techniques: Tokenization

The text is split into smaller units. We can use either sentence tokenization or word tokenization based on our problem statement. You can easily tokenize the sentences and words of the text with the tokenize module of NLTK.



```
import nltk
nltk.download('punkt'
)
from nltk.tokenize import word_tokenize, sent_tokenize

print(word_tokenize('اهلا ومرحبا بكم في معسكر علم البيانات '))
print(sent_tokenize(' اهلا ومرحبا بكم في معسكر علم البيانات، اليوم هو بداية البرمجة اللغوية
العصبية'))
print(word_tokenize('The weather is warm and sunny today'))
print(sent_tokenize("The weather is warm and sunny. It is a great day to go to the
```

```
Output:
['اهلا ' ، 'و' ، 'مرحبا' ، 'بكم' ، 'في' ، 'معسكر' ، 'علم' ، 'البيانات']
['اهلا ومرحبا بكم في معسكر علم البيانات'، ' اليوم هو بداية البرمجة اللغوية العصبية']
['The', ‘weather', 'is', 'warm', 'and', 'sunny', 'today']
['The weather is warm and sunny.', 'It is a great day to go to the beach']
```

text Representation Techniques: N-Grams

N-grams are contiguous sequences of nn items (words, characters, etc.) extracted from a text. They capture the context an patterns in the text by considering combinations of nn adjacent elements.

Types of N-grams:

1. **Unigrams (1-grams):** Single items from the text.
   - **Input:** "Machine learning is exciting."
   - **Output:** ["Machine", "learning", "is", "exciting"]
2. **Bigrams (2-grams):** Pairs of adjacent items.
   - **Input:** "Machine learning is exciting."
   - **Output:** ["Machine learning", "learning is", "is exciting"]
3. **Trigrams (3-grams):** Triplets of adjacent items.
   - **Input:** "Machine learning is exciting."
   - **Output:** ["Machine learning is", "learning is exciting"]

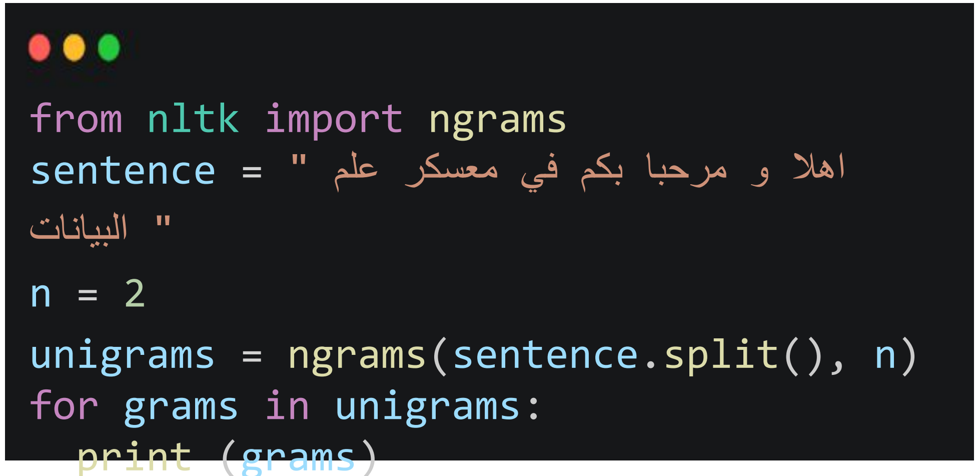
Importance:

- **Context Capture:** Provides context by considering the relationship between adjacent items.
- **Pattern Recognition:** Helps in identifying patterns and relationships within the text.
- **Feature Engineering:** Useful in text classification, language modeling, and other NLP tasks.

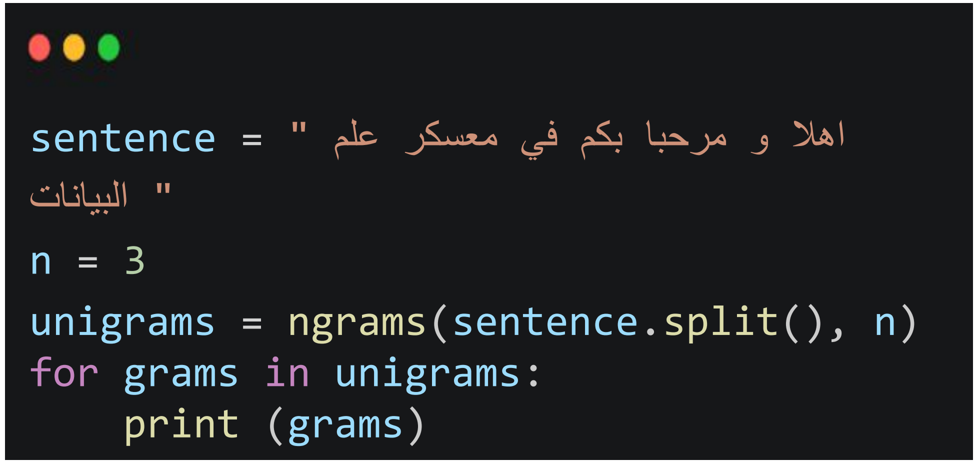
Text Preprocessing Techniques: N-Grams

N-grams are continuous sequences of words or symbols, or tokens in a document. In technical terms, they are a type of tokenization which can be defined as the neighboring sequences of items in a document.

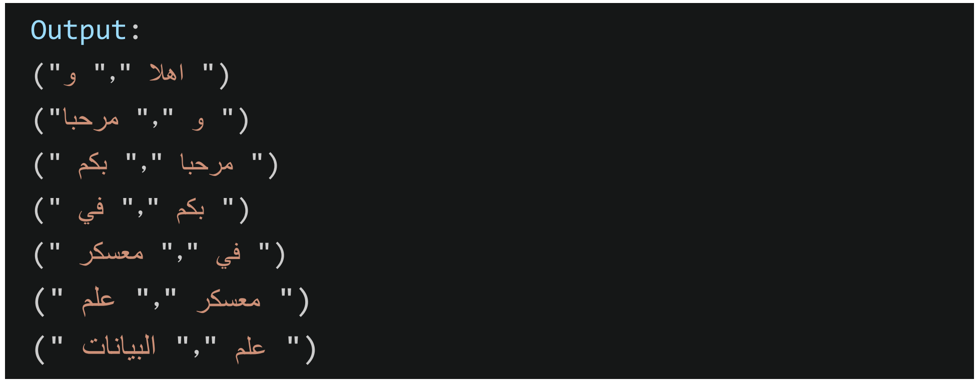
Arabic Example:



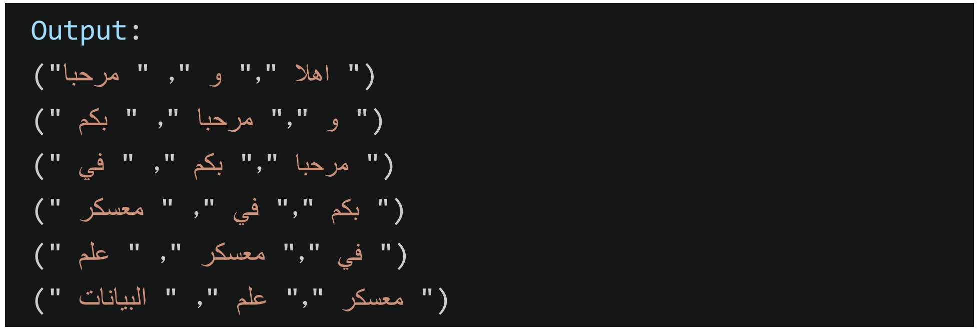
```
from nltk import ngrams
sentence = " اهلا و مرحبا بكم في معسكر علم البيانات "
n = 2
unigrams = ngrams(sentence.split(), n)
for grams in unigrams:
  print (grams)
```



```
sentence = " اهلا و مرحبا بكم في معسكر علم البيانات "
n = 3
unigrams = ngrams(sentence.split(), n)
for grams in unigrams:
    print (grams)
```



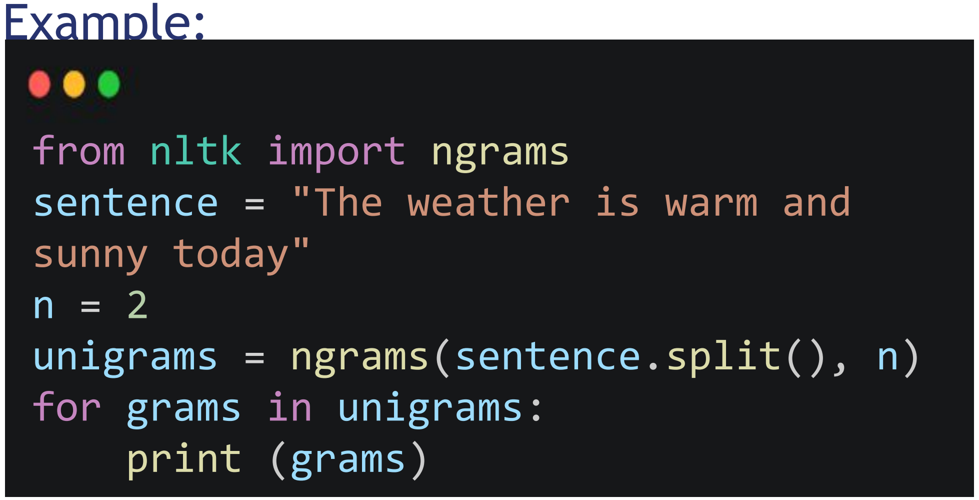
```
Output:
(" اهلا "," و")
(" و "," مرحبا")
(" مرحبا "," بكم ")
(" بكم "," في ")
(" في "," معسكر ")
(" معسكر "," علم ")
(" علم "," البيانات ")
```



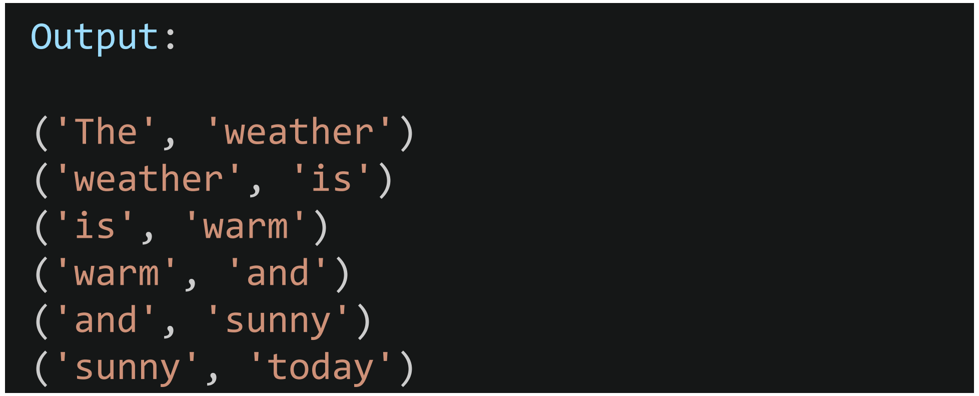
```
Output:
(" اهلا "," و " ," مرحبا")
(" و "," مرحبا " ," بكم ")
(" مرحبا "," بكم " ," في ")
(" بكم "," في " ," معسكر ")
(" في "," معسكر " ," علم ")
(" معسكر "," علم " ," البيانات ")
```

Text Preprocessing Techniques: N-Grams

English Example:

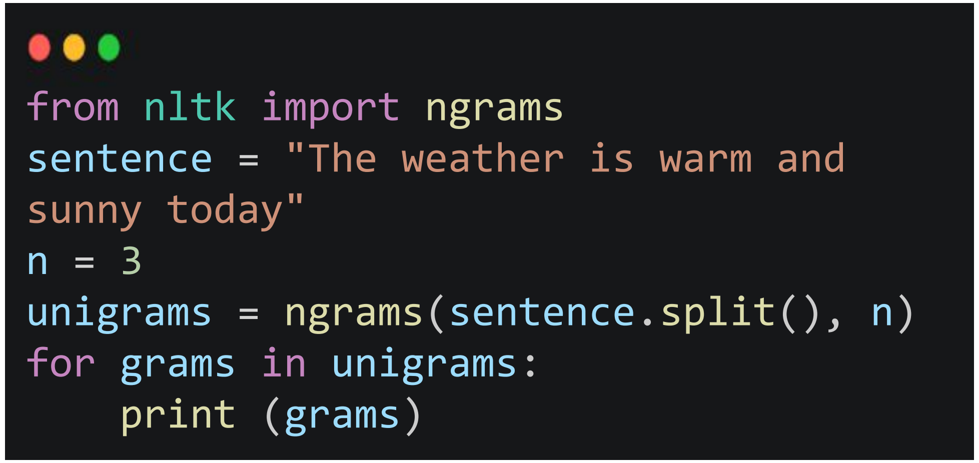


```
from nltk import ngrams
sentence = "The weather is warm and
sunny today"
n = 2
unigrams = ngrams(sentence.split(), n)
for grams in unigrams:
    print (grams)
```

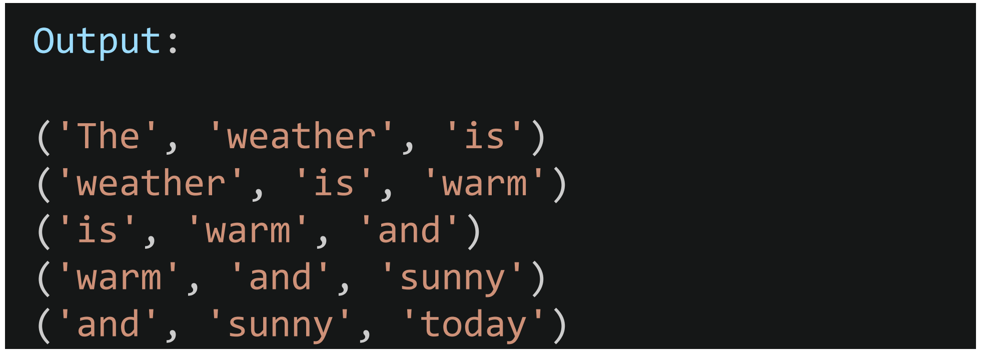


```
Output:

('The', 'weather')
('weather', 'is')
('is', 'warm')
('warm', 'and')
('and', 'sunny')
('sunny', 'today')
```



```
from nltk import ngrams
sentence = "The weather is warm and
sunny today"
n = 3
unigrams = ngrams(sentence.split(), n)
for grams in unigrams:
    print (grams)
```



```
Output:

('The', 'weather', 'is')
('weather', 'is', 'warm')
('is', 'warm', 'and')
('warm', 'and', 'sunny')
('and', 'sunny', 'today')
```

Conclusion

- NLU is a subset of NLP that focuses on machine comprehension of language. By deciphering the intent behind words, NLU enables more accurate and context-aware interactions between humans and AI.

Tutorial

7-Natural Language Processing/1-Introduction to Natural Language Processing (NLP)LAB/NLU.ipynb

Natural language generation (NLG)

What is Natural Language Generation?

- is a software process driven by artificial intelligence that produces natural written or spoken language from structured and unstructured data. It helps computers to feed back to users in human language that they can comprehend, rather than in a way a computer might.
- Converts both structured and unstructured data into human-like language.

Source

NLG Process

- Text Planning

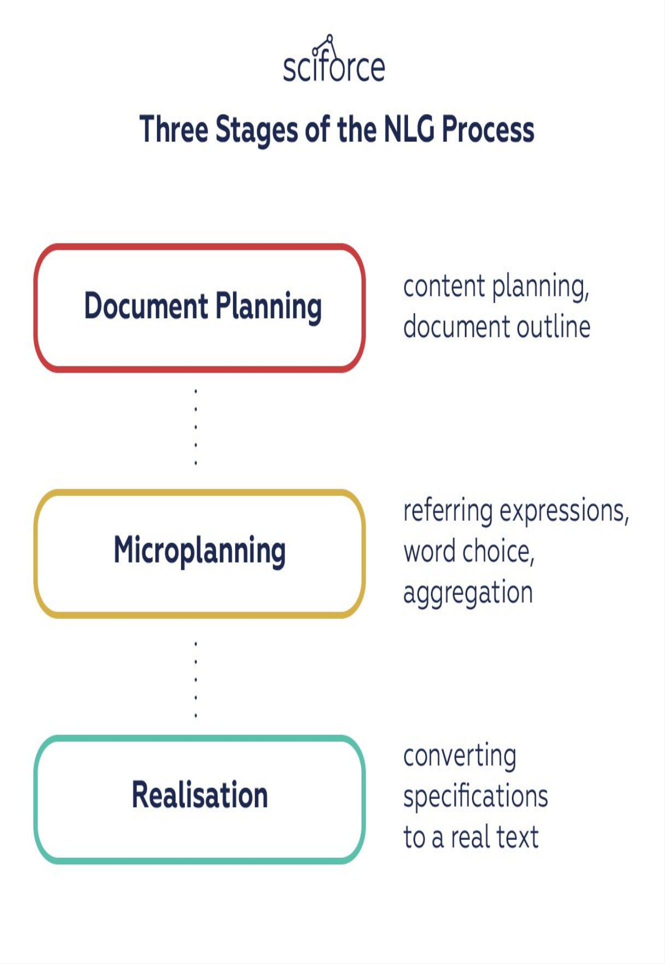
Decides what to say and in what order.

- Microplanning

Referring expressions.

- Realisation

Converts the structured plan into natural, grammatically correct text.

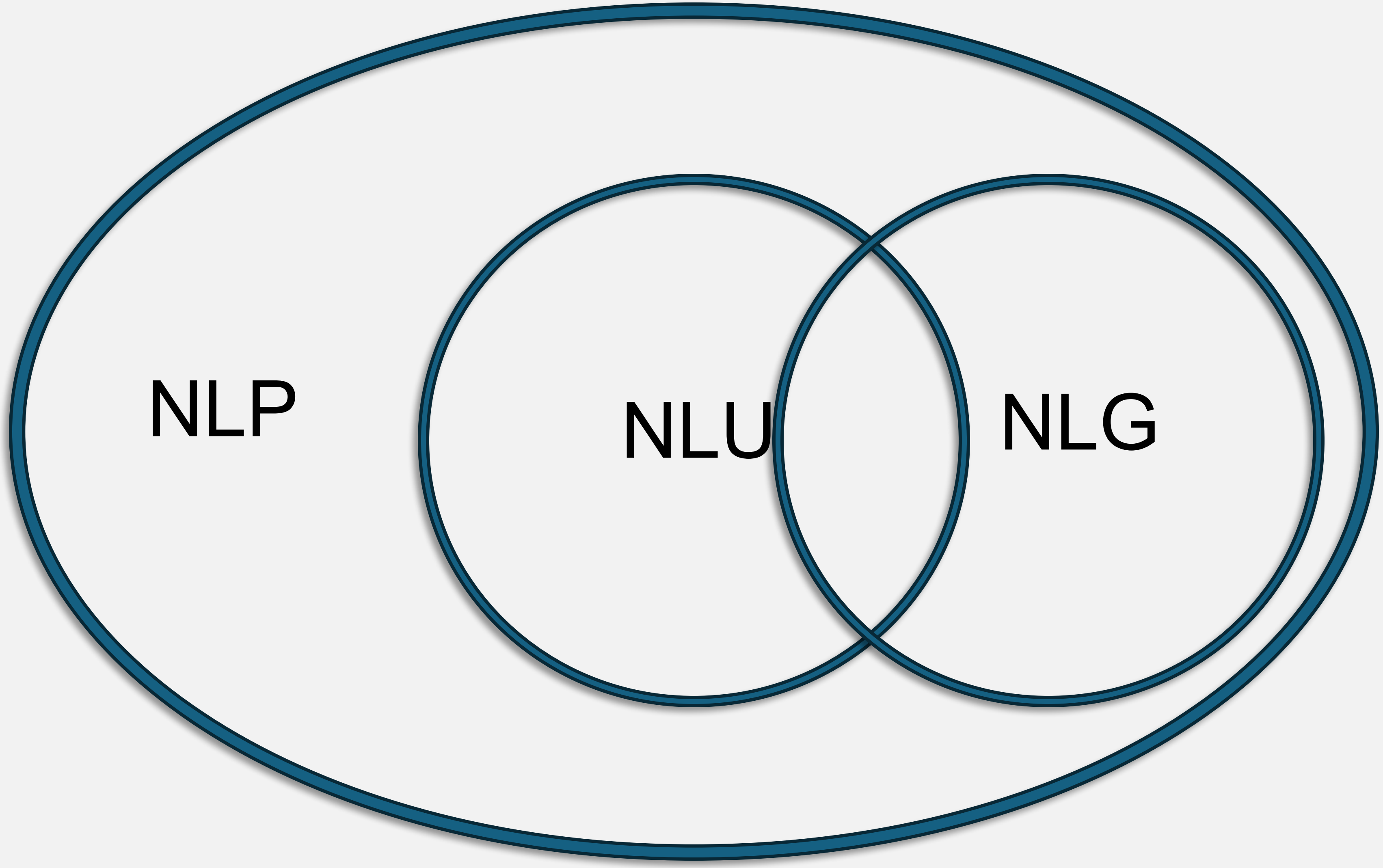

sciforce
Three Stages of the NLG Process
Document Planning
content planning, document outline
Microplanning
referring expressions, word choice, aggregation
Realisation
converting specifications to a real text

NLG Applications

- Surface Realization

Converts the structured plan into natural, grammatically correct text.

How NLP, NLU, NLG Correspond



NLP vs. NLU vs. NLG

So, as far as we have already examined the core of a trio called NLP NLG NLU, let's now try to find out the main difference between NLP NLU and NLG.

NLP	NLU	NLG
It encompasses the techniques and processes for analyzing, understanding, and deriving meaning from human language. NLP deals with both the syntax (structure) and semantics (meaning) of the text.	A subfield of NLP focused on comprehending the meaning and context of language. NLU involves interpreting the intent behind the text, extracting relevant information, and making decisions based on that understanding.	It involves creating coherent and contextually relevant human-like text from structured data. NLG is used to generate responses, summaries, or descriptions that are natural and understandable to humans.

Source

NLP vs. NLU vs. NLG

Feature	**NLP**	**NLU**	**NLG**
Input	Text data / Speech	Text data / Speech	Structured data/ Instructions
Output	Converts unstructured data to structured data	Reads unstructured data	Generates unstructured data
Focus	Processing and analyzing language data	Interpreting and understand language input	Producing coherent text or speech
Application	• Text Analysis • Smart Assistance • Language Translation	• Speech Recognition • Sentiment Analysis	• Chat Bots • Virtual Assistants

Source

Conclusion

- NLG is the flip side of NLU, where machines generate human-like text. NLG applications include content creation, report generation, and more, making it a powerful tool for automating communication.

Thank you!

SDAIA
الهيئة السعودية للبيانات
والذكاء الاصطناعي
Saudi Data & AI Authority