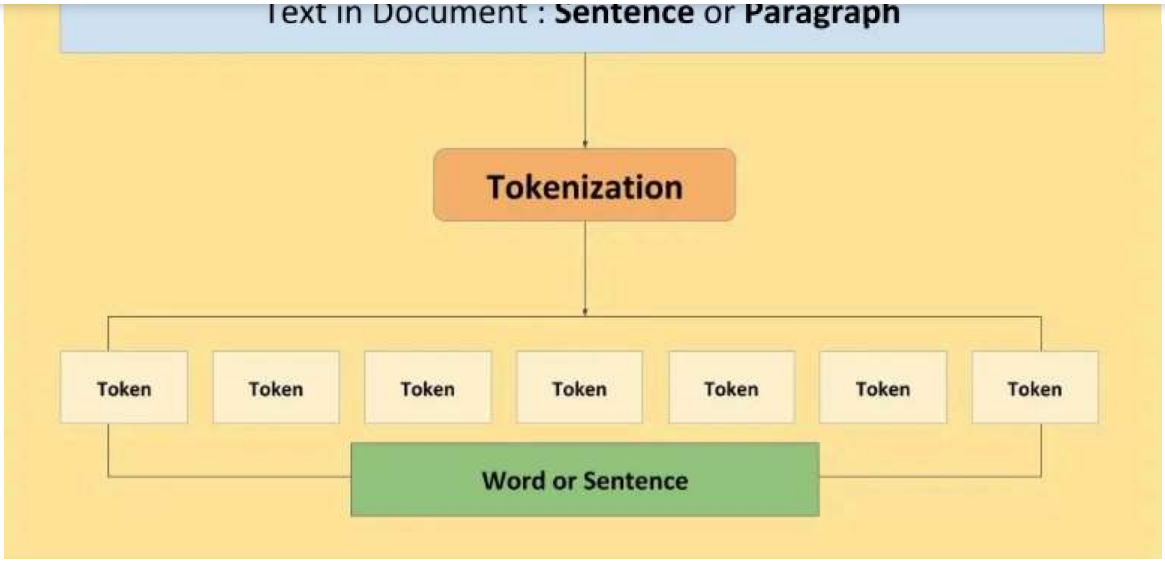
Aysel Aydin

Summary

Tokenization is a foundational process in NLP that involves breaking

3 — Tokenization in NLP: The Art of Breaking Down Text Data



In this article, we will cover the **Tokenization** topic. Before we start, I recommend you read the 2 articles I have previously covered about text preprocessing.

- 1 — Text Preprocessing Techniques for NLP
- 2 — Stemming & Lemmatization in NLP: Text Preprocessing Techniques

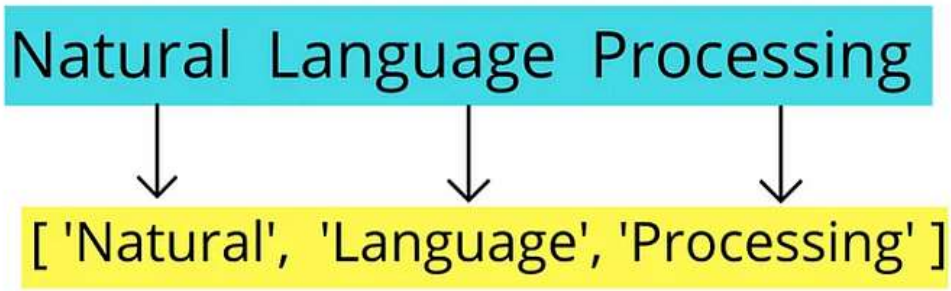
What is Tokenization?

Tokenization is one of the most common tasks when it comes to working with text data. It is the process of breaking a sentence or text into individual words or subwords, known as tokens.

plays a crucial role in understanding the structure and meaning of the text.

Why is Tokenization Essential for NLP?

Let's discuss the importance of tokenization when analyzing social media comments through text analysis.



Imagine a company that wants to monitor comments posted on social media platforms about its products and services. These comments contain valuable information about customer satisfaction, product quality and potential issues. However, these comments are often written in complex, lengthy and sometimes with language errors.

Here's how tokenization plays a crucial role in this scenario:

Understanding the Text: Social media comments are typically long and complex. Tokenization helps break down these comments into words and

tokens: “I am” and “very satisfied.”

Sentiment Analysis: The company aims to understand customer satisfaction. Tokenization can help identify positive or negative expressions. For example, the phrase “I had a great experience” indicates a positive sentiment because of the presence of the word “great.”

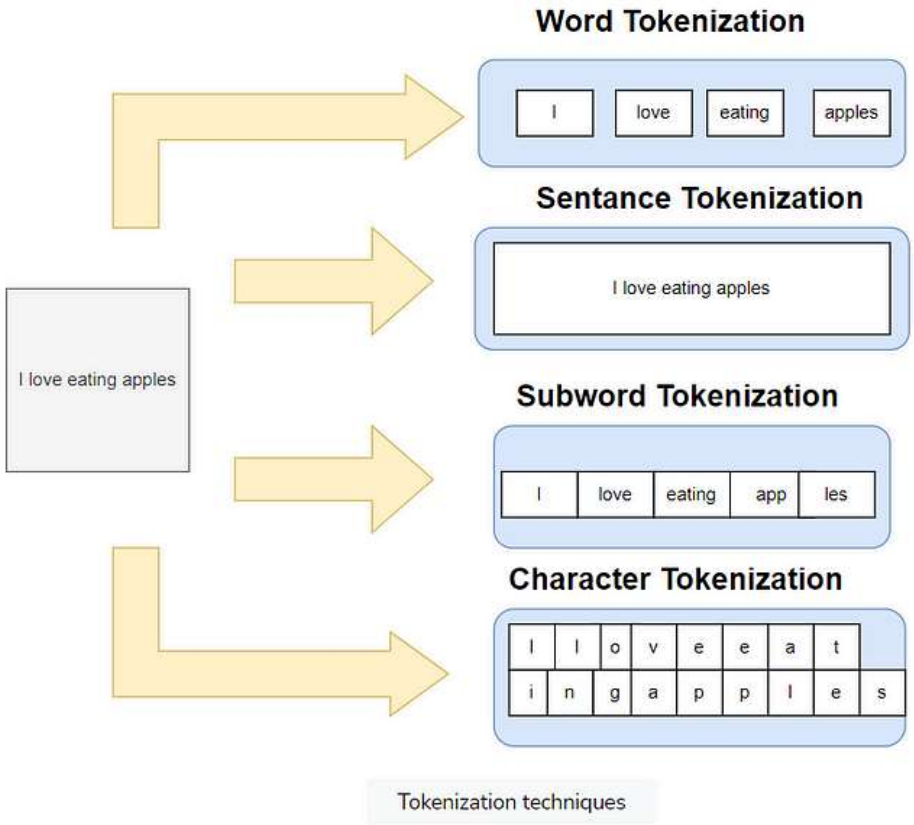
Word Frequency: Tokenization can be used to calculate the frequency of specific words. By understanding which words are used most frequently, the company can identify key topics related to their product or service.

Text Classification: Categorizing comments into specific categories or sentiments is essential. For instance, the company might want to analyze comments related to a particular product separately. Tokenization assists in classifying comments into these categories.

In summary, tokenization is a fundamental step in NLP, and it is essential for understanding and extracting valuable insights from complex text data like social media comments. It enables the company to analyze and make informed decisions based on customer feedback and sentiments. This example illustrates how tokenization is crucial in real-life NLP applications to process, understand and analyze text data effectively.

techniques.

How does tokenization work in NLP?



Tokenization techniques

There are different methods and libraries available to perform tokenization. NLTK, Gensim and Keras are some of the libraries that can be used to accomplish the task.

and the same separation done for sentences is called **sentence tokenization.**

Word Tokenization

```
import nltk
from nltk.tokenize import word_tokenize

text = "In this article, we are learning word tokenization using NLTK."

tokens = word_tokenize(text)
print(tokens)
```

```
Output:
['In', 'this', 'article', ',', 'we', 'are', 'learning', 'word', 'tokenization
```

Sentence Tokenization

> Firstly, install the NLTK library and download Punkt tokenizer models if you haven't already.

```
nltk.download('punkt')
```

After the installation, let's continue with the sentence tokenization code.

```
import nltk
from nltk.tokenize import sent_tokenize

text = "Hello! Sentence tokenization is essential for breaking down a text in
 its constituent sentences, which is a fundamental step in natural language
 processing. It allows you to work with sentences individually,
 making it easier to perform tasks like sentiment analysis, text summarizatio
 and machine translation. NLTK provides a simple way to achieve sentence
 tokenization in Python."

sentences = sent_tokenize(text)

for sentence in sentences:
    print(sentence)
```

```
Output:
Hello!
Sentence tokenization is essential for breaking down a text into its constitu
It allows you to work with sentences individually, making it easier to perfor
NLTK provides a simple way to achieve sentence tokenization in Python.
```

```
text = "Hello World!"

characters = list(text)

print("Characters:", characters)
```

```
Output:
Characters: ['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '!']
```

You can also perform these operations using spaCy, Keras and Gensim. When I add it to Github, I will add the link here.

> I will cover the subject of “N-gram tokenization” in more detail in another article.

Conclusion

Through this article, we have learned about different tokenizers from NLTK.

In summary, tokenization is a critical preprocessing step in many NLP tasks. It is fundamental to NLP as it transforms raw text data into a format that can

meaningful information and patterns from text data.

I hope it will be a useful article for you. If you stayed with me until the end, thank you for reading! Happy coding ✊

Contact Accounts: Twitter, LinkedIn

NLP Tokenization Word Tokenization Nltk Token

Recommended from ReadMedium


Aysel Aydin

8—Label Encoding & One-hot Encoding

In this article, we will talk about label encoding and one hot encoding, their usage areas and differences.

3 min read


Rahul Kumar

Classification 2. Token Classification 3...

3 min read

Eastgate Software

What is Tokenization in NLP? Everything You Need to Understand

Tokenization is a foundational concept in Natural Language Processing (NLP), a branch of artificial intelligence that enables machines to...

6 min read

Jo Wang

Deep Learning Part 5 -How to prevent overfitting

Techniques used to prevent overfitting in deep learning models:

4 min read

Mdabdullahalhasib

A Complete Guide of Output Parser with LangChain Implementation

Explore how we can get output from the LLM model into any structural format like CSV, JSON, or others, and create your custom parser also.

6 min read

pritesh