Text Representation

Text Representation is the process of converting **unstructured** textual data into a **structured** format so that it can be used for computational analysis. It is a **fundamental** step in **Natural Language Processing (NLP)**, where text is prepared for tasks like **sentiment analysis, text classification, information retrieval, and machine translation.**

There are several **key techniques** used in text representation, including **Tokenization, Stemming, Stopword Removal, Named Entity Recognition (NER), and N-Gram Modeling.** Let's go through each in detail.

---

1. Tokenization

Tokenization is the process of breaking text into **smaller meaningful units** called **tokens**. These tokens could be **words, sentences, or even characters** depending on the type of tokenization used.

Types of Tokenization:

1. **Word Tokenization** – Splits a sentence into words.
2. **Sentence Tokenization** – Splits a paragraph into sentences.
3. **Character Tokenization** – Splits a word into individual characters.

Example 1: Word Tokenization

Input:
"Natural Language Processing is amazing!"

Output:
```
["Natural", "Language", "Processing", "is", "amazing", "!"]
```

Example 2: Sentence Tokenization

Input:
"NLP is a subfield of AI. It helps machines understand text."

Output:
```
["NLP is a subfield of AI.", "It helps machines understand text."]
```

Example 3: Character Tokenization

Input:
"AI"

Output:
`["A", "I"]`

👉 **Use Cases:** Tokenization is **fundamental** in NLP tasks such as **text classification, chatbot responses, and search engines**.

---

2. Stemming

Stemming is a process that reduces a word to its **root or base** form by removing prefixes and suffixes. This helps **normalize words** and reduce vocabulary size.

Common Stemming Algorithms

1. **Porter Stemmer** – Removes common endings like **"-ing", "-ed", "-es"**.
2. **Snowball Stemmer** – An advanced version of Porter Stemmer.
3. **Lancaster Stemmer** – More aggressive than Porter Stemmer.

Example 1: Stemming Using Porter Stemmer

Input Words: `"running", "runner", "ran"`
Output: `"run"`

Word	Stemmed Output
Running	Run
Runner	Run
Ran	Run

👉 **Use Cases:** Stemming is useful for **search engines** where different word forms of the same meaning should be treated as one.

---

3. Stopword Removal

Stopwords are commonly used words in a language (like **"is", "the", "and", "in"**) that **do not add much meaning** and are often removed in NLP tasks.

Example of Stopword Removal

Input Sentence:
"Text mining is the process of extracting useful information from text."

After Stopword Removal:
`["Text", "mining", "process", "extracting", "useful", "information"]`

👉 **Use Cases:**

1. Improves **text analysis** by focusing only on important words.
2. Helps in **search engines** to retrieve better results.

---

4. Named Entity Recognition (NER)

Named Entity Recognition (NER) is used to identify **real-world entities** like **names, locations, organizations, dates, and more**.

Example of NER

Input Sentence:
"Elon Musk is the CEO of Tesla, which is based in the USA."

Output:

- **"Elon Musk" → Person**
- **"Tesla" → Organization**
- **"USA" → Location**

👉 **Use Cases:**

1. **Chatbots** (Extracting user-specific information)
2. **News classification** (Identifying important entities)
3. **Search engines** (Enhancing query understanding)

---

5. N-Gram Modeling

N-Gram is a sequence of **N words** used for **predictive text analysis** and **language modeling**.

Types of N-Grams:

1. **Unigram (n=1):** Individual words
    - Example: `"I love NLP"` → `["I", "love", "NLP"]`
2. **Bigram (n=2):** Two-word combinations
    - Example: `"I love NLP"` → `["I love", "love NLP"]`
3. **Trigram (n=3):** Three-word combinations
    - Example: `"I love NLP"` → `["I love NLP"]`

Example:

Input Sentence:
"Natural Language Processing is powerful."

Bigram Representation:
`["Natural Language", "Language Processing", "Processing is", "is powerful"]`

Trigram Representation:
`["Natural Language Processing", "Language Processing is", "Processing is powerful"]`

👉 **Use Cases:**

1. **Text Prediction** (e.g., mobile keyboards predicting next words).
2. **Machine Translation** (e.g., Google Translate uses N-Grams for language modeling).
3. **Speech Recognition** (e.g., converting speech to text more accurately).

---

6. Workflow of Text Representation

The **pipeline** for processing text in NLP generally follows these steps:

1️⃣ **Tokenization** → Breaks text into words or sentences.
2️⃣ **Stopword Removal** → Removes unimportant words.
3️⃣ **Stemming/Lemmatization** → Reduces words to root form.
4️⃣ **Named Entity Recognition (NER)** → Identifies names, places, and organizations.
5️⃣ **N-Gram Modeling** → Analyzes patterns of words.

---

7. Applications of Text Representation

1️⃣ Search Engines

- Google uses **Tokenization, Stopword Removal, and N-Grams** to improve search results.

2️⃣ Chatbots

- Virtual assistants like **Siri, Alexa, Google Assistant** rely on **NER, Tokenization, and N-Grams** for understanding queries.

3️⃣ Sentiment Analysis

- Businesses use **Text Representation** to analyze customer reviews and classify them as **positive, negative, or neutral**.

4️⃣ Text Generation

- AI systems like **GPT (ChatGPT)** use **N-Gram Modeling and NLP** for generating human-like text.