Text Preprocessing – Core Steps (Detailed Guide)

Text preprocessing is a fundamental step in Natural Language Processing (NLP). Since machines cannot understand raw human language directly, preprocessing is used to clean, normalize, and structure text data before it is passed to NLP models.

1. Lowercasing (Case Normalization)

Lowercasing converts all characters in text to lowercase to ensure uniformity. Without this step, words like 'NLP' and 'nlp' are treated as different tokens.

Example:
'Machine Learning Is FUN' → 'machine learning is fun'

Lowercasing should be avoided in tasks like Named Entity Recognition (NER), where capitalization provides important meaning.

2. Removing Punctuation and Special Characters

Punctuation and special characters usually do not contribute to the semantic meaning of text and may add noise.

Example:
'Hello!!! How are you?' → 'Hello How are you'

However, punctuation may be important in sentiment analysis or emotional text.

3. Removing Numbers

Numbers are removed when they do not add meaningful information to the text.

Example:
'Order number 12345 delivered' → 'Order number delivered'

Numbers should be retained in domains such as finance, healthcare, or analytics.

4. Stopword Removal

Stopwords are commonly used words such as 'is', 'the', 'and', which usually carry little meaning.

Example:
'This is a very good movie' → 'good movie'

Stopword removal helps reduce dimensionality and improves model efficiency, but it should be avoided in sentiment analysis and question-answering systems.

5. Stemming

Stemming reduces words to their root form by removing suffixes.

Example:
'running' → 'run'
'played' → 'play'

Stemming is fast but may produce invalid or incomplete words.

6. Lemmatization

Lemmatization converts words into their meaningful base form using linguistic rules.

Example:
'better' → 'good'
'running' → 'run'

Lemmatization is more accurate than stemming but computationally slower.

7. Removing Extra Whitespaces

Extra spaces, tabs, and line breaks are removed to maintain consistent formatting.

Example:
'Hello World' → 'Hello World'

8. Handling Contractions

Contractions are expanded into their full form for better clarity.

Example:
"can't" → "cannot"
"don't" → "do not"

9. Handling Emojis and Symbols

Emojis can either be removed or converted into text depending on the use case.

Example:
'I love NLP ■' → 'I love NLP happy'

Emoji handling is especially useful in sentiment analysis and social media data.

Complete Text Preprocessing Pipeline Example

Raw Text: "I CAN'T believe NLP is AWESOME!!! ■■" After Preprocessing: "cannot believe nlp awesome" This example demonstrates how preprocessing transforms noisy raw text into clean, machine-readable input.

Conclusion: Text preprocessing is a crucial step that directly impacts the performance of NLP models. Proper preprocessing improves accuracy, reduces noise, and enhances model efficiency.