# Day 3: Tokenization in Urdu & Pashto NLP

Sami Uddin Shinwari
Subject Specialist – IT | NLP Researcher
MS Data Science (FAST NU Peshawar)
BS Computer Science (UET Peshawar)

January 15, 2026

# What is Tokenization?

- Tokenization is the process of breaking text into small units called **tokens**
- Tokens can be words, subwords, characters, phrases, or numbers
- It is the **first step** of every NLP pipeline
- Tokenization makes text understandable for machines

# Why Tokenization is Tricky

- Token definitions depend on language and use case
- Punctuation can split words incorrectly
- Example:

*She isn't at the house.*

Naive tokens: **isn** | **t**

**Issue:** Apostrophes break meaningful words.

# Why Whitespace Works in English

- English uses spaces consistently
- Word boundaries are usually clear

*This is a sentence*

**Tokens:** This | is | a | sentence

# Why Whitespace Fails in Urdu

- Space is not a reliable word boundary
- Writers may omit or add spaces

**Urdu Examples:**

space) (no غلطالفاظ

space) (extra غ لط الفاظ

**Problem:** Both confuse tokenizers.

# Urdu Sentence Example

یہ ایک جملہ ہے

**Whitespace Tokens:**
یہ|ایک|جملہ|ہے

**Issue:** Compound words break easily in real data.

# Pashto Tokenization Challenges

- Arabic-based script like Urdu
- Inconsistent spacing in informal text
- Low-resource language → limited tools

**Pashto Example:**

دايوهبيلګهده

**Correct Form:**

دا يوه بيلګه ده

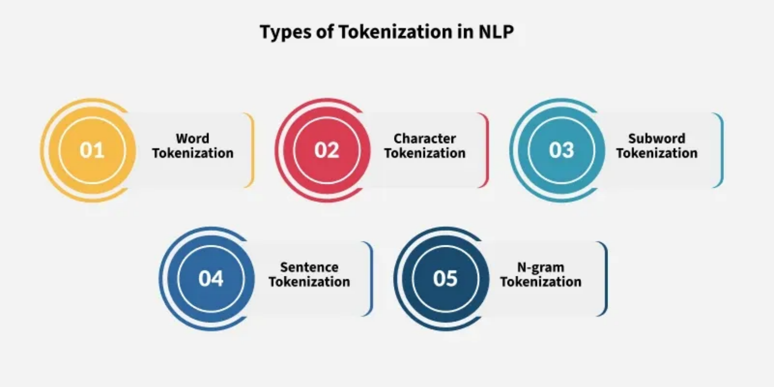
# Tokenization vs Chunking

- **Tokenization:** Splits text into words, subwords, or characters
- **Chunking:** Splits large text into manageable segments
- Chunking usually happens **before tokenization**
- LLMs tokenize only the most relevant chunks

# Types of Tokenization

- Character Tokenization
- Word Tokenization
- Phrase Tokenization
- Sentence Tokenization
- Subword Tokenization (BPE, SentencePiece)
- Number Tokenization

# Types of Tokenization (Visual)



Types of Tokenization in NLP

01 Word Tokenization

02 Character Tokenization

03 Subword Tokenization

04 Sentence Tokenization

05 N-gram Tokenization

# Why Tokenization Matters

- Wrong tokens → wrong POS tags
- NER performance drops
- TF-IDF and embeddings become noisy
- Tokenization errors propagate to all NLP tasks

# Professional Uses of Tokenization

- Information retrieval (search engines)
- Text preparation and feature extraction
- Sentiment analysis
- Generative AI and chatbots
- User feedback analysis

# Tokenization using NLTK (Python)

**Word Tokenization Example**

```
import nltk
from nltk.tokenize import word_tokenize

text = "Natural Language Processing is amazing!"
tokens = word_tokenize(text)
print(tokens)
```

**Output:**

```
['Natural', 'Language', 'Processing', 'is', 'amazing', '!']
```

# Tokenization using spaCy

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Tokenization is the first NLP step.")

for token in doc:
    print(token.text)
```

**Key Feature:** Context-aware tokenization

# Tokenization using Hugging Face

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
tokens = tokenizer.tokenize("Tokenization handles OOV words")

print(tokens)
```

**Uses:** BERT, GPT, T5, Transformer models

# Why Subword Tokenization?

- Handles Out-Of-Vocabulary (OOV) words
- Reduces vocabulary size
- Preserves word structure
- Used in modern NLP models

Example: *unusual* → un | usual

# Why Tokenization Matters

- Affects POS tagging accuracy
- Impacts NER and embeddings
- Poor tokenization = poor NLP results

# Key Takeaways

- Whitespace tokenization is NOT enough
- Urdu & Pashto need language-aware tokenizers
- Subword tokenization works best for low-resource languages
- Tokenization quality defines NLP success

**#Day3 #Tokenization #UrduNLP #PashtoNLP**

# References

- Rehman et al. (2011). Challenges in Urdu Tokenization. ACL.
- Becker & Riaz (2012). Urdu Morphology Study.
- Khan et al. (2023). Pashto Text Processing Challenges.