Transformer Tokenization

How AI Models Understand Text

HKBU FIN7830, 2025

Shengwei YOU

Press Space for next page →

What is Tokenization?

Tokenization is the process of breaking down text into smaller pieces that AI models can understand.

Think of it like:

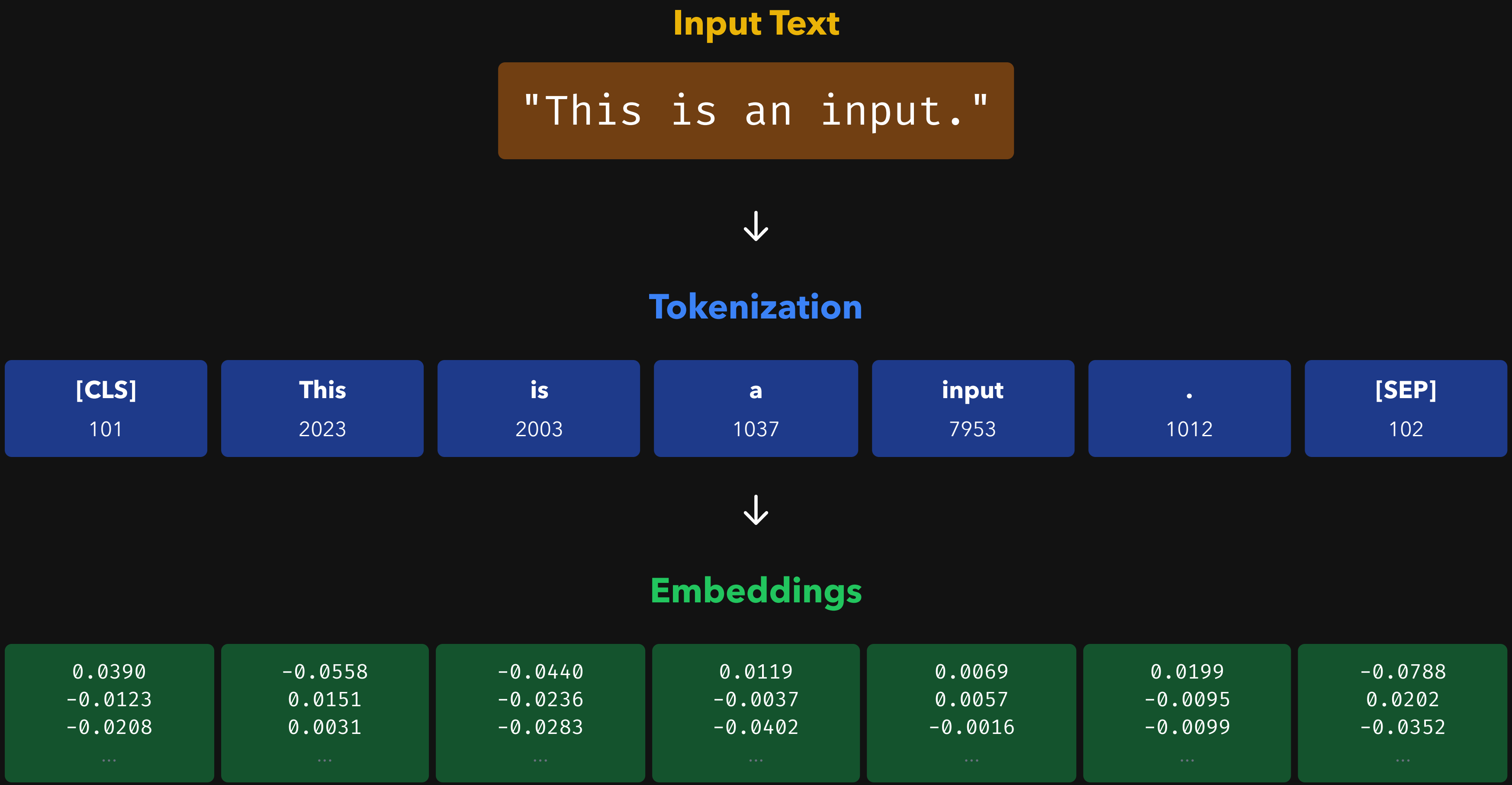
- 📝 Breaking a sentence into words
- 🔢 Converting words into numbers
- 🤖 Making text readable for computers

"I love Dogs!"

↓

['i', 'love', 'dogs', '!']

The Complete Pipeline: From Text to Embeddings



Input Text

```
"This is an input."
```

↓

Tokenization

↓

Embeddings

[CLS]	This	is	a	input	.	[SEP]
101	2023	2003	1037	7953	1012	102
0.0390 -0.0123 -0.0208 ...	-0.0558 0.0151 0.0031 ...	-0.0440 -0.0236 -0.0283 ...	0.0119 -0.0037 -0.0402 ...	0.0069 0.0057 -0.0016 ...	0.0199 -0.0095 -0.0099 ...	-0.0788 0.0202 -0.0352 ...

Step-by-Step: Tokenization Process

Original Sentence

"I love Dogs!"

After Tokenization

['i', 'love', 'dogs', '!']

What happens:

- ✅ Text is split into tokens (words/subwords)
- ✅ Converted to lowercase
- ✅ Punctuation becomes separate tokens

Token IDs: The Language of Numbers

Each token gets a unique number

Tokens

'i' → 1045

'love' → 2293

'dogs' → 5055

'!' → 999

Token IDs

```
[1045, 2293, 5055, 999]
```

These numbers are what the AI model actually processes!

Special Tokens: The Secret Helpers

When tokenizing, AI models add special tokens that provide structure:

[CLS]

Classification Token

- Placed at the beginning
- Represents the whole sentence
- Used for understanding context

[SEP]

Separator Token

- Marks the end of sentence
- Separates multiple sentences
- Signals completion

```
[CLS] i love dogs! [SEP]
```

Complete Example

Original:

```
"I love Dogs!"
```

After Tokenization:

```
['i', 'love', 'dogs', '!']
```

With Special Tokens:

```
[CLS] i love dogs! [SEP]
```

Token IDs:

```
[101, 1045, 2293, 5055, 999, 102]
```

Live Demo: Tokenization in Python

Let's see tokenization in action with Python code:

```python
# Import the DistilBERT tokenizer
from transformers import DistilBertTokenizer

# Initialize the tokenizer
tokenizer = DistilBertTokenizer.from_pretrained('distilbert-base-uncased')

# Our example sentence
sentence = "I love Dogs!"

# Step 1: Tokenize (split into tokens)
tokens = tokenizer.tokenize(sentence)
print(f"Tokens: {tokens}")

# Step 2: Convert to Token IDs
token_ids = tokenizer.encode(sentence, add_special_tokens=True)
print(f"\nToken IDs: {token_ids}")

# Step 3: Decode back to see special tokens
decoded = tokenizer.decode(token_ids)
print(f"\nDecoded with special tokens: {decoded}")
```

Try it yourself! Edit the sentence variable and see how different text gets tokenized.

Why Does This Matter?

🎯 Purpose

Tokenization converts human language into a format that AI models can process and understand

💡 Key Points

- Text → Tokens → Numbers
- Special tokens add structure
- Every model uses this process

Tokenization is the first step in making AI understand human language!

Summary

Tokenization breaks text into smaller pieces

Each piece becomes a **number** (Token ID)

Special tokens like [CLS] and [SEP] add structure

This process enables AI to **understand** and **process** text

Thank you! 🎓