Hugging Face Tokenizers

Lesson Downloads

Show TranscriptSummarize Video

HuggingFace tokenizers help us break down text into smaller, manageable pieces called tokens. These tokenizers are easy to use and also remarkably fast due to their use of the Rust programming language.

Technical Terms Explained:

Tokenization: It's like cutting a sentence into individual pieces, such as words or characters, to make it easier to analyze.

Tokens: These are the pieces you get after cutting up text during tokenization, kind of like individual Lego blocks that can be words, parts of words, or even single letters. These tokens are converted to numerical values for models to understand.

Pre-trained Model: This is a ready-made model that has been previously taught with a lot of data.

Uncased: This means that the model treats uppercase and lowercase letters as the same.

Code Example

```
from transformers import BertTokenizer

# Initialize the tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

# See how many tokens are in the vocabulary
tokenizer.vocab_size
# 30522
```

```
# Tokenize the sentence
tokens = tokenizer.tokenize("I heart Generative AI")

# Print the tokens
print(tokens)
# ['i', 'heart', 'genera', '##tive', 'ai']

# Show the token ids assigned to each token
print(tokenizer.convert_tokens_to_ids(tokens))
# [1045, 2540, 11416, 6024, 9932]
```

Resources

[Hugging Face Tokenizers documentation index](#)

[Hugging Face Tokenizer documentation](#)

[Hugging Face BertTokenizer documentation](#)