🤗   🔍 Search models, datasets, users...   ☰

Join the Hugging Face community

and get access to the augmented documentation experience

Sign Up   to get started

🔴 **Pytorch**   🌊 TensorFlow   💬 Ask a question
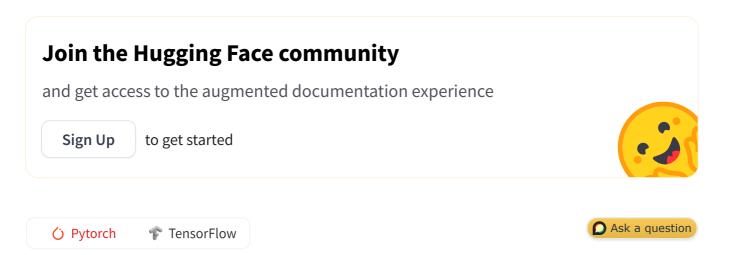
End-of-chapter quiz

1. What is the order of the language modeling pipeline?

☐   First, the model, which handles text and returns raw predictions. The tokenizer then makes sense of these predictions and converts them back to text when needed.

☐   First, the tokenizer, which handles text and returns IDs. The model handles these IDs and outputs a prediction, which can be some text.

☑   The tokenizer handles text and returns IDs. The model handles these IDs and outputs a prediction. The tokenizer can then be used once again to convert these predictions back to some text.

> **Correct!** Correct! The tokenizer can be used for both tokenizing and de-tokenizing.

Submit   **You got all the answers!**

2. How many dimensions does the tensor output by the base Transformer model have, and what are they?

☐   2: The sequence length and the batch size

☐   2: The sequence length and the hidden size

☑   3: The sequence length, the batch size, and the hidden size

> **Correct!** Correct!

Submit   **You got all the answers!**

3. Which of the following is an example of subword tokenization?

☑ WordPiece

> **Correct!** Yes, that's one example of subword tokenization!

☐ Character-based tokenization

☐ Splitting on whitespace and punctuation

☑ BPE

> **Correct!** Yes, that's one example of subword tokenization!

☑ Unigram

> **Correct!** Yes, that's one example of subword tokenization!

☐ None of the above

Submit   **You got all the answers!**

4. What is a model head?

☐ A component of the base Transformer network that redirects tensors to their correct layers

☐ Also known as the self-attention mechanism, it adapts the representation of a token according to the other tokens of the sequence

☑ An additional component, usually made up of one or a few layers, to convert the transformer predictions to a task-specific output

> **Correct!** That's right. Adaptation heads, also known simply as heads, come up in different forms: language modeling heads, question answering heads, sequence classification heads...

Submit   **You got all the answers!**

5. What is an AutoModel?

☐ A model that automatically trains on your data

☑ An object that returns the correct architecture based on the checkpoint

> **Correct!** Exactly: the `AutoModel` only needs to know the checkpoint from which to initialize to return the correct architecture.

☐ A model that automatically detects the language used for its inputs to load the correct weights

Submit   **You got all the answers!**

6. What are the techniques to be aware of when batching sequences of different lengths together?

☐ Truncating

☐ Returning tensors

☐ Padding

☑ Attention masking

> **Correct!** Absolutely! Attention masks are of prime importance when handling sequences of different lengths. That's not the only technique to be aware of, however.

Submit   **You didn't select all the correct answers, there's more!**

7. What is the point of applying a SoftMax function to the logits output by a sequence classification model?

☐ It softens the logits so that they're more reliable.

☐ It applies a lower and upper bound so that they're understandable.

☑ The total sum of the output is then 1, resulting in a possible probabilistic interpretation.

> **Correct!** Correct! That's not the only reason we use a SoftMax function, though.

Submit   **You didn't select all the correct answers, there's more!**

8. What method is most of the tokenizer API centered around?

☐ encode, as it can encode text into IDs and IDs into predictions

☑ Calling the tokenizer object directly.

> **Correct!** Exactly! The __call__ method of the tokenizer is a very powerful method which can handle pretty much anything. It is also the method used to retrieve predictions from a model.

☐ pad

☐ tokenize

Submit   **You got all the answers!**

9. What does the result variable contain in this code sample?

```python
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
result = tokenizer.tokenize("Hello!")
```

☑ A list of strings, each string being a token

> **Correct!** Absolutely! Convert this to IDs, and send them to a model!

☐ A list of IDs

☐ A string containing all of the tokens

Submit   **You got all the answers!**

NLP Course documentation
End-of-chapter quiz ⌄   🔍

```python
from transformers import AutoTokenizer, AutoModel

tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
model = AutoModel.from_pretrained("gpt2")

encoded = tokenizer("Hey!", return_tensors="pt")
result = model(**encoded)
```

☐ No, it seems correct.

☑ The tokenizer and model should always be from the same checkpoint.

> **Correct!** Right!

☐ It's good practice to pad and truncate with the tokenizer as every input is a batch.

Submit   **You got all the answers!**

← Basic usage completed!   ✓ Complete Chapter