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.
- [x] 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! 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
- [x] 3: The sequence length, the batch size, and the hidden size

Correct! Nicely done!

Submit **You got all the answers!**

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

- [x] WordPiece

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

- [ ] Character-based tokenization
- [ ] Splitting on whitespace and punctuation
- [x] BPE

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

- [x] 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
- [x] 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

- [x] 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?

- [x] Truncating

Correct! Yes, truncation is a correct way of evening out sequences so that they fit in a rectangular shape. Is it the only one, though?

- [ ] Returning tensors
- [x] Padding

Correct! Yes, padding is a correct way of evening out sequences so that they fit in a rectangular shape. Is it the only one, though?

- [x] 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 got all the answers!**

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.
- [x] 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
- [x] 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?

```
from transformers import AutoTokenizer

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

- [x] 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!

10. Is there something wrong with the following code?

```
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.
- [x] 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!**

<> Update on GitHub

← Optimized Inference Deployment

Complete Chapter