Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

Next item →

1. In the lectures, what is the name of the layer used to generate the vocabulary? **1 / 1 point**

- ○ Tokenizer
- ◉ TextVectorization
- ○ TextTokenizer
- ○ WordTokenizer

Correct
That's right!

2. Once you have generated a vocabulary, how do you encode a string sentence to an integer sequence? **1 / 1 point**

- ○ Use the texts_to_sequences() method of the adapted TextVectorization layer.
- ◉ Pass the string to the adapted TextVectorization layer.
- ○ Use the texts_to_tokens() method of the adapted TextVectorization layer.
- ○ Pass the string to the get_vocabulary() method.

Correct
That's right!

3. If you have a number of sequences of different lengths, how do you ensure that they are understood when fed into a neural network? **1 / 1 point**

- ○ Process them on the input layer of the Neural Network using the pad_sequences property
- ◉ Use the pad_sequences function from tf.keras.utils
- ○ Make sure that they are all the same length using the pad_sequences method of the TextVectorization layer
- ○ Specify the input layer of the Neural Network to expect different sizes with dynamic_length

Correct
That's right!

4. What happens at encoding when passing a string that is not part of the vocabulary? **1 / 1 point**

- ○ The word isn't encoded, and the sequencing ends
- ○ The word is replaced by the most common token
- ○ The word isn't encoded, and is replaced by a zero in the sequence
- ◉ An out-of-vocabulary token is used to represent it.

Correct
Correct!

5. When padding sequences, if you want the padding to be at the end of the sequence, how do you do it? **1 / 1 point**

- ○ Call the padding method of the pad_sequences object, passing it 'post'
- ○ Pass padding='after' to pad_sequences when initializing it
- ○ Call the padding method of the pad_sequences object, passing it 'after'
- ◉ Pass padding='post' to pad_sequences when initializing it

Correct
That's right!

6. What's one way to convert a list of strings named 'sentences' to integer sequences? Assume you adapted a TextVectorization layer and assigned it to a variable named 'vectorize_layer'. **1 / 1 point**

- ◉ vectorize_layer(sentences)
- ○ vectorize_layer.fit_to_text(sentences)
- ○ vectorize_layer.tokenize(sentences)
- ○ vectorize_layer.fit(sentences)

Correct
Correct!

7. If you have a number of sequences of different length, and call pad_sequences on them, what's the default result? **1 / 1 point**

- ○ Nothing, they'll remain unchanged
- ◉ They'll get padded to the length of the longest sequence by adding zeros to the beginning of shorter ones
- ○ They'll get padded to the length of the longest sequence by adding zeros to the end of shorter ones
- ○ They'll get cropped to the length of the shortest sequence

Correct
Correct!

8. Using the default settings, how does the TextVectorization standardize the string inputs? **1 / 1 point**

- ○ By arranging the strings in alphabetical order.
- ○ By stripping punctuation.
- ○ By lowercasing the strings.
- ◉ By lowercasing and stripping punctuation.

Correct
That's right! This is the default setting in the 'standardize' parameter of the TextVectorization layer.