✓ **Congratulations! You passed!**
TO PASS 80% or higher

[Keep Learning]

GRADE
100%

Week 4 Quiz

LATEST SUBMISSION GRADE
100%

1. **What is the name of the method used to tokenize a list of sentences?**   `1 / 1 point`

   ○ tokenize(sentences)

   ○ tokenize_on_text(sentences)

   ○ fit_to_text(sentences)

   ● fit_on_texts(sentences)

   ✓ **Correct**

2. **If a sentence has 120 tokens in it, and a Conv1D with 128 filters with a Kernal size of 5 is passed over it, what's the output shape?**   `1 / 1 point`

   ○ (None, 120, 124)

   ● (None, 116, 128)

   ○ (None, 120, 128)

   ○ (None, 116, 124)

   ✓ **Correct**

3. **What is the purpose of the embedding dimension?**   `1 / 1 point`

   ● It is the number of dimensions for the vector representing the word encoding

   ○ It is the number of dimensions required to encode every word in the corpus

   ○ It is the number of letters in the word, denoting the size of the encoding

   ○ It is the number of words to encode in the embedding

   ✓ **Correct**

4. **IMDB Reviews are either positive or negative. What type of loss function should be used in this scenario?**   `1 / 1 point`

   ○ Binary Gradient descent

   ● Binary crossentropy

   ○ Adam

   ○ Categorical crossentropy

   ✓ **Correct**

5. **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

   ○ Make sure that they are all the same length using the pad_sequences method of the tokenizer

   ● Use the pad_sequences object from the tensorflow.keras.preprocessing.sequence namespace

   ○ Specify the input layer of the Neural Network to expect different sizes with dynamic_length

   ✓ **Correct**

6. **When predicting words to generate poetry, the more words predicted the more likely it will end up gibberish. Why?**   `1 / 1 point`

   ○ Because the probability of prediction compounds, and thus increases overall

   ○ Because you are more likely to hit words not in the training set

   ○ It doesn't, the likelihood of gibberish doesn't change

   ● Because the probability that each word matches an existing phrase goes down the more words you create

   ✓ **Correct**

7. **What is a major drawback of word-based training for text generation instead of character-based generation?**   `1 / 1 point`

   ○ There is no major drawback, it's always better to do word-based training

   ○ Word based generation is more accurate because there is a larger body of words to draw from

   ○ Character based generation is more accurate because there are less characters to predict

   ● Because there are far more words in a typical corpus than characters, it is much more memory intensive

   ✓ **Correct**

8. **How does an LSTM help understand meaning when words that qualify each other aren't necessarily beside each other in a sentence?**   `1 / 1 point`

   ○ They don't

   ○ They shuffle the words randomly

   ○ They load all words into a cell state

   ● Values from earlier words can be carried to later ones via a cell state

   ✓ **Correct**