**1.**

Question 1

What is the name of the object used to tokenize sentences?

1 point

TextTokenizer

CharacterTokenizer

Tokenizer

WordTokenizer

**2.**

Question 2

What is the name of the method used to tokenize a list of sentences?

1 point

tokenize\_on\_text(sentences)

tokenize(sentences)

fit\_to\_text(sentences)

fit\_on\_texts(sentences)

**3.**

Question 3

Once you have the corpus tokenized, what’s the method used to encode a list of sentences to use those tokens?

1 point

texts\_to\_sequences(sentences)

text\_to\_tokens(sentences)

texts\_to\_tokens(sentences)

text\_to\_sequences(sentences)

**4.**

Question 4

When initializing the tokenizer, how do you specify a token to use for unknown words?

1 point

oov\_token=<Token>

unknown\_token=<Token>

unknown\_word=<Token>

out\_of\_vocab=<Token>

**5.**

Question 5

If you don’t use a token for out of vocabulary words, what happens at encoding?

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 skipped in the sequence

The word isn’t encoded, and is replaced by a zero in the sequence

**6.**

Question 6

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 point

Process them on the input layer of the Neural Network using the pad\_sequences property

Specify the input layer of the Neural Network to expect different sizes with dynamic\_length

Make sure that they are all the same length using the pad\_sequences method of the tokenizer

Use the pad\_sequences function from the tensorflow.keras.preprocessing.sequence namespace

**7.**

Question 7

If you have a number of sequences of different length, and call pad\_sequences on them, what’s the default result?

1 point

They’ll get padded to the length of the longest sequence by adding zeros to the beginning of shorter ones

They’ll get cropped to the length of the shortest sequence

They’ll get padded to the length of the longest sequence by adding zeros to the end of shorter ones

Nothing, they’ll remain unchanged

**8.**

Question 8

When padding sequences, if you want the padding to be at the end of the sequence, how do you do it?

1 point

Call the padding method of the pad\_sequences object, passing it ‘post’

Call the padding method of the pad\_sequences object, passing it ‘after’

Pass padding=’post’ to pad\_sequences when initializing it

Pass padding=’after’ to pad\_sequences when initializing it

**1.**

Question 1

What is the name of the TensorFlow library containing common data that you can use to train and test neural networks?

1 point

TensorFlow Data

There is no library of common data sets, you have to use your own

TensorFlow Datasets

TensorFlow Data Libraries

**2.**

Question 2

How many reviews are there in the IMDB dataset and how are they split?

1 point

60,000 records, 80/20 train/test split

50,000 records, 80/20 train/test split

60,000 records, 50/50 train/test split

50,000 records, 50/50 train/test split

**3.**

Question 3

How are the labels for the IMDB dataset encoded?

1 point

Reviews encoded as a boolean true/false

Reviews encoded as a number 1-10

Reviews encoded as a number 0-1

Reviews encoded as a number 1-5

**4.**

Question 4

What is the purpose of the embedding dimension?

1 point

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

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

**5.**

Question 5

When tokenizing a corpus, what does the num\_words=n parameter do?

1 point

It specifies the maximum number of words to be tokenized, and picks the first ‘n’ words that were tokenized

It specifies the maximum number of words to be tokenized, and picks the most common ‘n-1’ words

It specifies the maximum number of words to be tokenized, and stops tokenizing when it reaches n

It errors out if there are more than n distinct words in the corpus

**6.**

Question 6

To use word embeddings in TensorFlow, in a sequential layer, what is the name of the class?

1 point

tf.keras.layers.Word2Vector

tf.keras.layers.WordEmbedding

tf.keras.layers.Embed

tf.keras.layers.Embedding

**7.**

Question 7

IMDB Reviews are either positive or negative. What type of loss function should be used in this scenario?

1 point

Adam

Binary Gradient descent

Categorical crossentropy

Binary crossentropy

**8.**

Question 8

When using IMDB Sub Words dataset, our results in classification were poor. Why?

1 point

Sequence becomes much more important when dealing with subwords, but we’re ignoring word positions

Our neural network didn’t have enough layers

We didn’t train long enough

The sub words make no sense, so can’t be classified

**1.**

Question 1

Why does sequence make a large difference when determining semantics of language?

0 / 1 point

It doesn’t

Because the order of words doesn’t matter

Because the order in which words appear dictate their impact on the meaning of the sentence

Because the order in which words appear dictate their meaning

Incorrect

Sorry, wrong answer.

**2.**

Question 2

How do Recurrent Neural Networks help you understand the impact of sequence on meaning?

1 / 1 point

They don’t

They carry meaning from one cell to the next

They look at the whole sentence at a time

They shuffle the words evenly

Correct

That's right!

**3.**

Question 3

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 shuffle the words randomly

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

They load all words into a cell state

They don’t

Correct

Correct!

**4.**

Question 4

What keras layer type allows LSTMs to look forward and backward in a sentence?

1 / 1 point

Unilateral

Bidirectional

Bilateral

Bothdirection

Correct

Correct!

**5.**

Question 5

What’s the output shape of a bidirectional LSTM layer with 64 units?

0 / 1 point

(128,None)

(None, 128)

(128,1)

(None, 64)

Incorrect

Sorry, wrong answer.

**6.**

Question 6

When stacking LSTMs, how do you instruct an LSTM to feed the next one in the sequence?

0 / 1 point

Do nothing, TensorFlow handles this automatically

Ensure that return\_sequences is set to True on all units

Ensure that return\_sequences is set to True only on units that feed to another LSTM

Ensure that they have the same number of units

Incorrect

Not quite.

**7.**

Question 7

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, 128)

(None, 120, 124)

(None, 116, 124)

(None, 116, 128)

Correct

That's right!

**8.**

Question 8

What’s the best way to avoid overfitting in NLP datasets?

1 / 1 point

Use LSTMs

Use GRUs

Use Conv1D

None of the above

**1.**

Question 1

When predicting words to generate poetry, the more words predicted the more likely it will end up gibberish. Why?

1 point

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

Because the probability of prediction compounds, and thus increases overall

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

**2.**

Question 2

What is a major drawback of word-based training for text generation instead of character-based generation?

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

**3.**

Question 3

What are the critical steps in preparing the input sequences for the prediction model?

1 point

Converting the seed text to a token sequence using texts\_to\_sequences.

Splitting the dataset into training and testing sentences.

Generating subphrases from each line using n\_gram\_sequences.

Pre-padding the subphrases sequences.

**4.**

Question 4

In natural language processing, predicting the next item in a sequence is a classification problem. Therefore, after creating inputs and labels from the subphrases, we one-hot encode the labels. What function do we use to create one-hot encoded arrays of the labels?

1 point

tf.keras.utils.img\_to\_array

tf.keras.utils.SequenceEnqueuer

tf.keras.utils.to\_categorical

tf.keras.preprocessing.text.one\_hot

**5.**

Question 5

True or False: When building the model, we use a sigmoid activated Dense output layer with one neuron per word that lights up when we predict a given word.

1 point

True

False