

✔ **Congratulations! You passed!**

Grade received **100%** To pass 80% or higher

Go to next item

Week 4 Quiz

Latest Submission Grade 100%

1. 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 that each word matches an existing phrase goes down the more words you create
- ☐ It doesn't, the likelihood of gibberish doesn't change
- ☐ Because the probability of prediction compounds, and thus increases overall
- ☐ Because you are more likely to hit words not in the training set

✔ **Correct**
That's right!

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

1 / 1 point

- ☐ 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
- ☐ Word based generation is more accurate because there is a larger body of words to draw from
- ☐ There is no major drawback, it's always better to do word-based training

✔ **Correct**
Correct!

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

1 / 1 point

- ☐ Splitting the dataset into training and testing sentences.
- ☒ Generating subphrases from each line using `n_gram_sequences`.

✔ **Correct**
Keep it up!

- ☒ Pre-padding the subphrases sequences.

✔ **Correct**
You've got it!

- ☐ Converting the seed text to a token sequence using `texts_to_sequences`.

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 / 1 point

- ☒ `tf.keras.utils.to_categorical`
- ☐ `tf.keras.utils.SequenceEnqueuer`

- ☐ tf.keras.utils.img_to_array
- ☐ tf.keras.preprocessing.text.one_hot

✓ **Correct**
Nailed it!

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 / 1 point

- ☒ False
- ☐ True

✓ **Correct**
Absolutely!