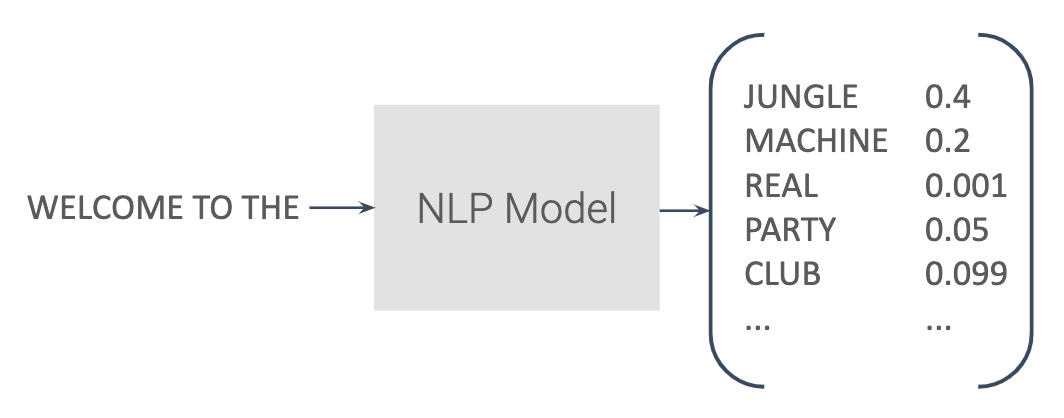
TEXT GENERATION

Text generation can be done through simply predicting the next most likely word, given an input sequence. This can be done over and over by feeding the original input sequence, plus the newly predicted end word, as the next input sequence to the model. As such, the full output generated from a very short original input can effectively go on however long you want it to be.

The only real change to the network here is the output layer will now be equivalent to a node per each possible new word to generate - so, if you have 1,000 possible words in your corpus, you’d have an output array of length 1,000. You’ll also need to change the loss function from binary cross-entropy to categorical cross entropy - before, we had only a 0 or 1 as output, now there are potentially thousands of output “classes” (each possible word).



Text Generation takes an input and outputs probabilities for the next most probable word.

**Constructing a Text Generation Model**

As noted before, there are hardly any differences in the model itself, other than changing the number of nodes in the output layer and changing the loss function.

The more obvious changes come in working with the input and output data. The input data takes chunks of sequences and just splits off the final word as its label. So, if we had the sentence “I went to the beach with my dog”, and we had a max input length of five words, we’d get:

*Input:****I went to the beach***

*Label:****with***

Now, that’s not the only sequence that will come from the sentence! We would also get:

*Input:****went to the beach with***

*Label:****my***

And:

*Input:****to the beach with my***

*Label:****dog***

That’s how the N-Grams used in the pre-processing work - a single input sequence might actually become a series of sequences and labels.

With the output of the network, I’ll let you mostly investigate that code in the Colab on the next page, but the important thing there is that you can keep looping and creating more text by just appending the next word onto the previous input sequence.