What do you think would be the next character? It may be a space, or a punctuation symbol (if the word was), or it could be a if the word was . The more characters we have in the sequence, the easier it is for us to predict what could be a reasonable next character.

A recurrent neural network can potentially work with and remember a very large number of characters in a sequence; this requires a huge amount of processing and some advanced math to understand what is going on (I'm not sure I'm there yet!). We are going to use a limited number of characters, n, and try to guess the $n+1$th character using simpler probability model.

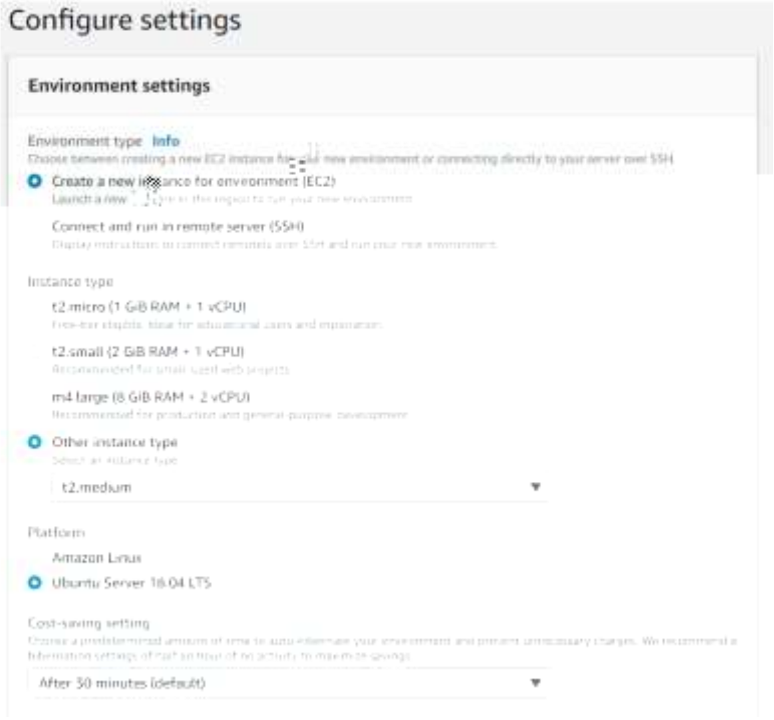
Let's say you have a lot of Shakespeare's texts. And you want to know the probability of a given character being the next character after a sequence of n characters. Let's call this the probability of character c, given that sequence s of n characters just occurred, or $P(c|s)$. The most intuitive way is to find all the occurrences of s in Shakespeare's text, and count the number of times the different characters appear, dividing by the total number of occurrences of s. This is called a maximum likelihood estimate. If a given character never appears after a sequence, it will have a probability of 0.

Once you have the probabilities, you will generate the next character in a sequence. But you can't only use the character with the maximum likelihood of appearing, or else your text will always be the same given that same starting sequence. Instead you will choose a random character based on the distribution. Note that we will force the first characters to always be the first characters in the original Shakespeare text.

There are three parts to this project.

- first, create your Cloud9 development environment
- then you will train your model and update a local development Redis data store; this is written in a console application. Redis is an in-memory data store that is much quicker than a relational database for certain types of data:

    simple data with a key-value model, where the keys are strings



Configure settings

Environment settings

Environment type Info
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH

● Create a new instance for environment (EC2)
Launch a new instance in this region to run your new environment

Connect and run in remote server (SSH)
Display instructions to connect remotely over SSH and run your new environment.

Instance type

t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

t2.small (2 GiB RAM + 1 vCPU)
Recommended for small-sized web projects.

m4.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.

● Other instance type
Select an instance type.

t2.medium ▼

Platform

Amazon Linux

● Ubuntu Server 18.04 LTS

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default) ▼

You can use either Amazon Linux or Ubuntu, the main differences from our perspective being that Amazon Linux uses and Ubuntu uses as package managers. The instructions below assume Ubuntu on Cloud9.

Remember to update the software on the new Cloud9 environment:

Your instructor has created a new repository (project in GitLab terminology) for the assignment. Set your git configuration and clone the repo created for you:

```
                          Your Name
                          you
```

Edit the README.md file.

And finally, you may want to go into preferences (the gear icon to the right), scroll down, and stop your environment after an hour (so that you don't have to keep logging back in).

You will write a script, split into functions and/or classes, which is invoked from the console. It will iteratively read the file and populate an array and then write to a local Redis data store.

We need to install Redis on Cloud9. First we will install Composer, the PHP package manager:

Next, install Predis, which is the PHP Redis client:

The associative array of arrays will look like this (partially), for example:

Substrings are keys		Values are arrays with character key and frequency value	
“aya”	→	‘ ’	→

$redis -