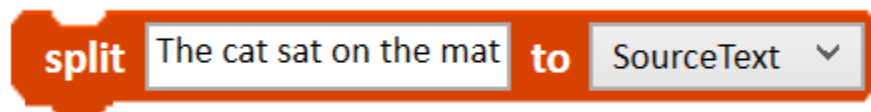# The cat sat on the cat

Program a robot to speak using probability

Most humans learn speech instinctually. Robots on the other hand have to be trained and there are several different ways to do so.

One of the ways to teach a robot to talk is with a Markov chain. By feeding the robot a source text it learns the probability of which word will follow another. By this method a robot can generate text of its own, using the source text as its vocabulary. Let's try out a simple example.

<u>Load the program</u>

This block establishes "The cat sat on the mat" as the source text.

It splits the text up into single words and adds them to a list called SourceText.

when space key pressed
set seedword to the
say seedword until done
split The cat sat on the mat to SourceText

This group of blocks sets the variable, seedword, and then instructs the robot to say the seed word.

What would you hear the robot say?
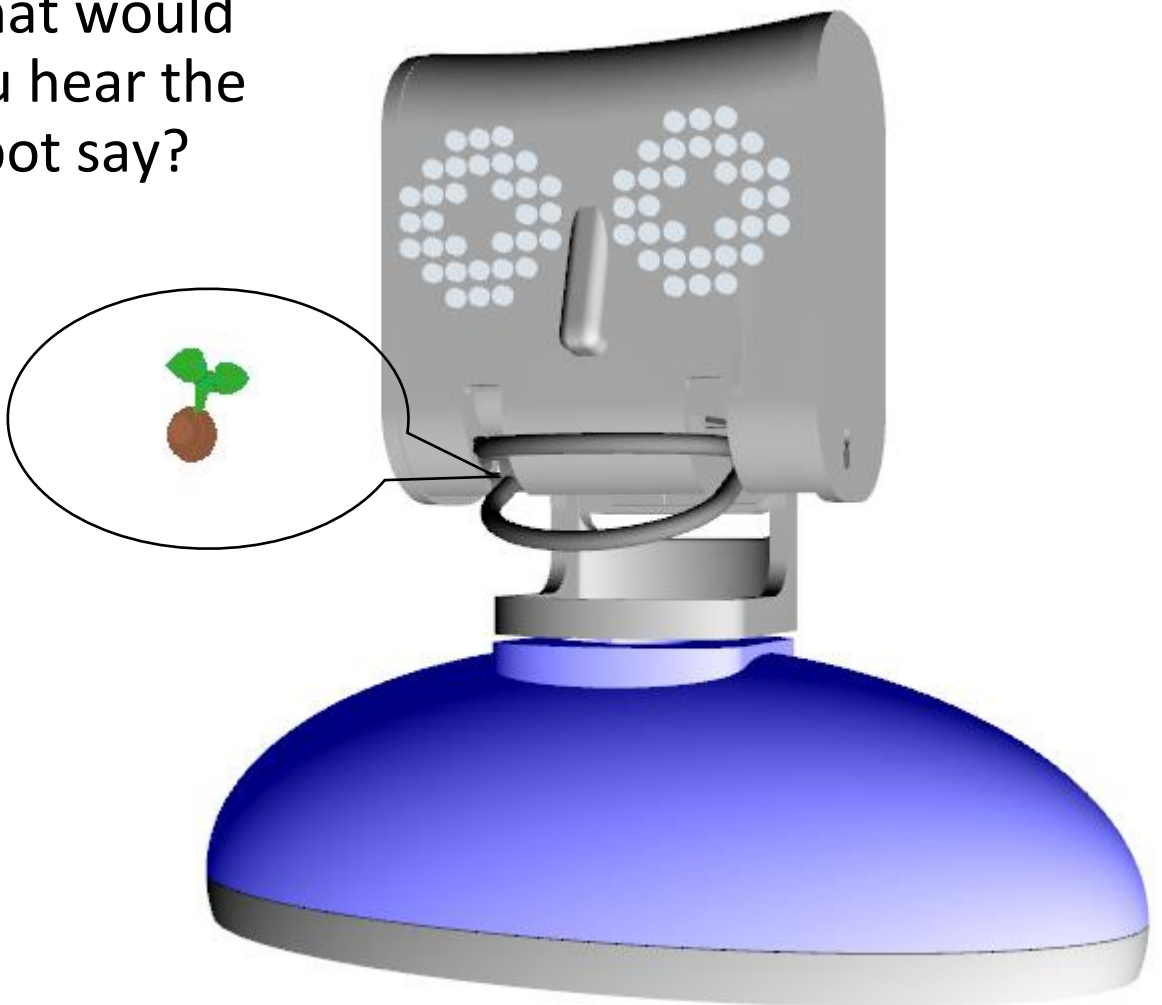
In our example the robot would say "the". You can set the seedword variable to anything and the robot will say it. But if you want the robot to keep talking then the seed word must be one the words in the SourceText list.

Create a variable called nextwordposition and set it to 1.

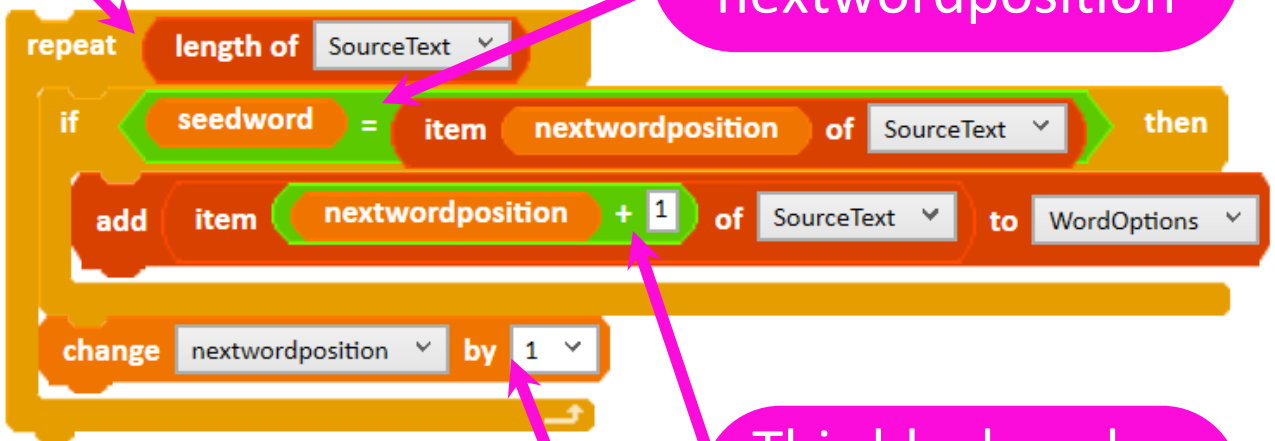This number refers to a position in the SourceText

We will use this variable to give the robot a reference point in the source text so they know where we want them to look.

Using an If, Then block we can tell the robot to search the SourceText list for the seed word and whenever it occurs add the word that follows it to a list called WordOptions.

This block makes sure the robot searches the whole source text

This operator block asks the robot whether the seed word matches the word at nextwordposition

```
repeat    length of  SourceText ⌄
  if    seedword = item  nextwordposition  of  SourceText ⌄   then
      add   item   nextwordposition + 1   of  SourceText ⌄  to  WordOptions ⌄
  change  nextwordposition ⌄  by  1 ⌄
```

This block tells the robot to move on to the next word. It will then see if the new word matches the seed word

This block makes the robot add the word after the seed word match to the WordOptions list

Once the robot has searched the whole of the text and added words to the WordOptions list, it's time for them to generate some text.

At the start of the program we had the robot say the seed word that we chose. Now that we have trained the robot on a source text they can choose the next word autonomously.

A new seed word is chosen at random from the WordOptions list.

```
set  seedword  to  item  random  of  WordOptions
say  seedword  until done
delete  all  of  WordOptions
```

The WordOptions list has to be deleted because it contains the words that followed the old seed word

```
when  space ▾  key pressed
set  seedword ▾  to  the ▾
say  seedword  until done
split  The cat sat on the mat  to  SourceText ▾
forever
    set  nextwordposition ▾  to  1 ▾
    repeat  length of  SourceText ▾
        if  seedword  =  item  nextwordposition  of  SourceText ▾  then
            add  item  nextwordposition  +  1  of  SourceText ▾  to  WordOptions ▾

        change  nextwordposition ▾  by  1 ▾

    set  seedword ▾  to  item  random ▾  of  WordOptions ▾
    say  seedword  until done
    delete  all ▾  of  WordOptions ▾
```

The program returns to the start of the forever block and begins the search for matches again with the new seed word
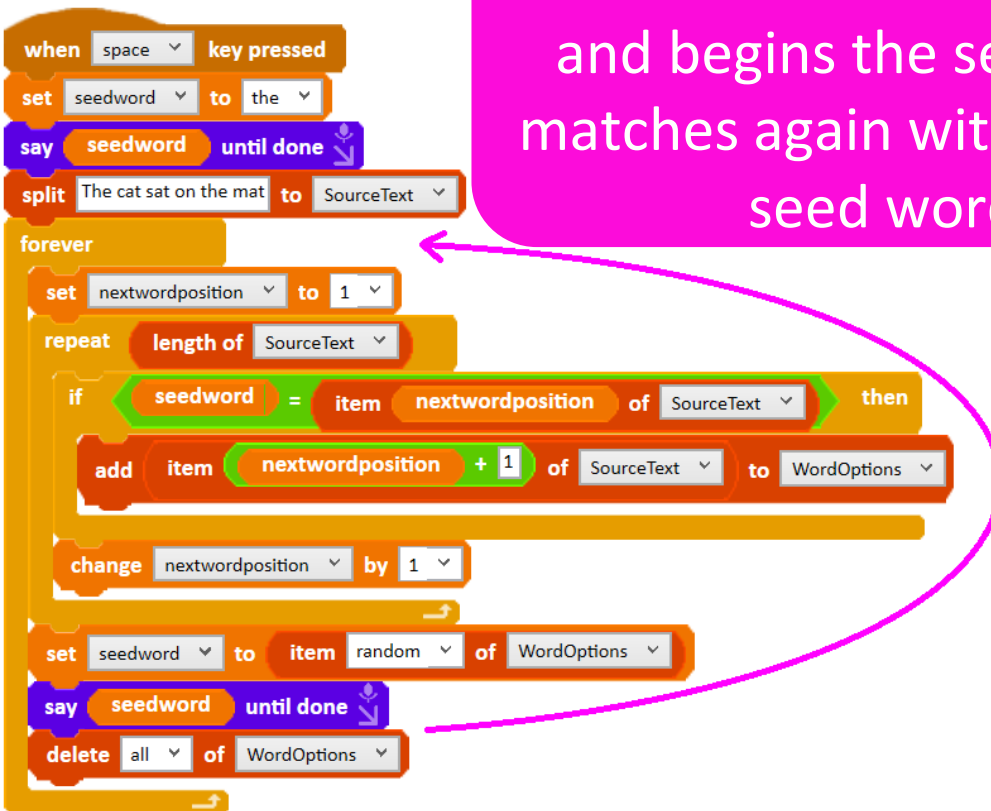
If the seed word appears multiple times in the source text and the words after it are different then the following words have a smaller probability of being chosen.

In our example the word "the" appears twice and the words that follow it are "cat" and "mat".

When the robot generates their text what is the chance of "cat" being chosen from the WordOptions list to follow "the"?

This is what our Markov chain looks like.

0.5

0.5

THE

CAT

MAT

1.0

1.0

ON

SAT

1.0
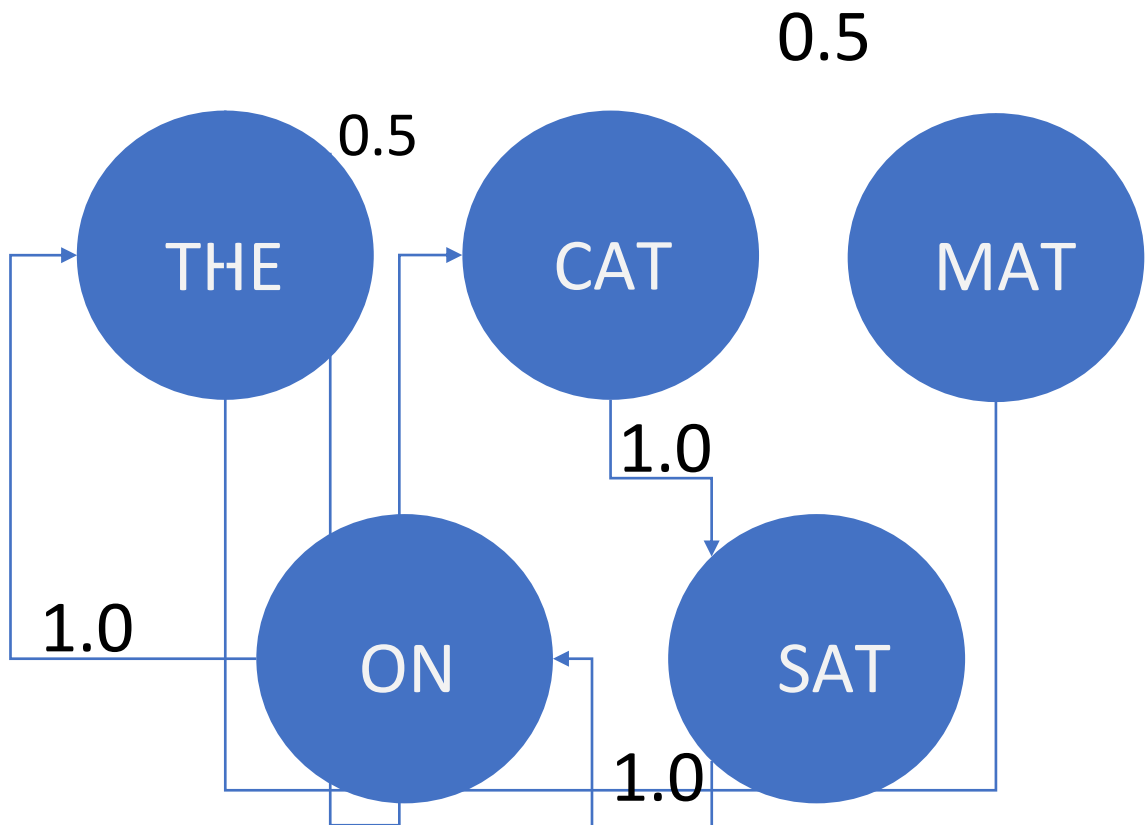
Both "cat" and "mat" have a 50% chance of being chosen by the robot to follow "the".

What happens if we set the original seed word to "mat"?

Can you find a way to make the program continue after "mat"?

Why not try using your own source text?

How do you think the length and content of the source text alters the text the robot generates?

The cat sat on the cat sat on the cat sat on the…