Which randomly generated Markov model uses a single letter to predict the next?

- Order zero
- Order-one

Correct

- Order-two
- Order-three

Continue

Introduction

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:03
Hello, in this module you're gonna learn about designing practical Java programs and software design and engineering skills in the context of a useful and amusing program. You'll develop and extend programs that use Markov Text Generation to train on text and then develop random texts based on that training data.

0:24
Although recognizing text using Markov techniques uses statistical and probabilistic methods, generating text can be done simply using the process and design we'll explain. Many of these ideas were made clear by a mathematician and computer scientist named Claude Shannon, who is often recognized as one of the principle thinkers behind what's called Information Theory.

0:47
This will allow you to practice Java programming and design concepts in the context of a program with very practical applications.

0:57
For example, both Google's page rank algorithm and many artificial intelligence machine learning algorithms rely on the same Markov concepts we'll explore.

1:08
The ideas we'll use have a simple model. Suppose you or a monkey pressed keys at random on a typewriter or keyboard, what text would be generated? Likely it would be nonsense. However, what if that keyboard had keys based on a training text? So there were more e's if the training text was English for example. The text might not be so random if there were 20 e keys for every z key on the keyboard.

1:38
We could extend this model so that if the a key was pressed, a new keyboard is used, one that had more t's than b's because there are many more words that start with the two letters at than the two letters ab. The second letter pressed would be based on probabilities as obtained from the training text and the first letter.

1:59
We could bring in a third keyboard based on pressing two letters. For example, if we press t, followed by h, then e would be very likely, r would be somewhat likely, and z would be extremely unlikely. We'll use these ideas in the Markov text generating programs you'll learn about. Let's look at the output of the early programs you'll develop.

2:22
Generating text based on training data.

2:26
In this case, the training data is a speech the German Chancellor Angela Merkel delivered to the European Parliament on October 7th, 2015.

2:36
An Order Zero text is based solely on the distribution of characters in the original text. There are lots of e's for example because that's a very common character. The space character is common too, but the text doesn't look like real text. An Order One text uses one letter to predict the next, so that the letter n is likely to follow an a, for example, but less likely to follow a p. The words you see are sometimes pronounceable, but they're not really words. Words like pen, woplits, and brarnt.

3:11
An Order Two text uses two characters to predict the third. This means that an e would be very likely to follow th, and a would be somewhat less likely to follow th. And a g would be very unlikely, because the sequence thg doesn't occur often in text.

3:29
You can see lots of words in sequences that are word-like, such as red, werty, hand, and unitions.

3:39
Although we're generating text, you might be able to recognize text and discover the training data. This is how Markov text recognition works.

3:48
For example, current spam algorithms often relay on Bayesian probability and Markov processes as part of recognizing spam based on training data.

3:59
Here's an Order Zero text. Can you recognize that perhaps the text isn't English based simply on the characters that appear? The accents on some of the letters make it possibly French.

4:11
In the Order One text, where one character is used to predict or generate the next character, it seems as though the text really might be French.

4:20
In the Order Two text, it seems very likely French. Here two characters predict the third.

4:28
In the Order Three text, you may even recognize La Marseillaise, the French National Anthem. You'll see in the programs you write that generating Order Three and generally Order N text is something that you could do. Here's an overview of what you'll learn in this module. You'll generalize a concept and an algorithm for generating random text using a Markov process. You'll see code for 0-order and 1-order text generation, and you'll generalize that to any, or N-order. You'll develop a Java Interface that allows the different concepts to be captured practically in code.

5:08
You'll capture these Abstractions in Java by developing and using interfaces.

5:14
That will help you design and test your program, an important part of building experience as a programmer and software engineer.

5:22
You'll also see that interfaces allow you to make program efficient without changing to many things at once, but separating design and implementation to facilitate efficiency.

5:34
You'll practice both software design and engineering skills while you develop these programs. This will give you practical and transferable experience to other domains of programming.

Downloads

- **Lecture Video** mp4
- **Subtitles (English)** WebVTT
- **Transcript (English)** txt

Would you like to help us translate the transcript and subtitles into additional languages?