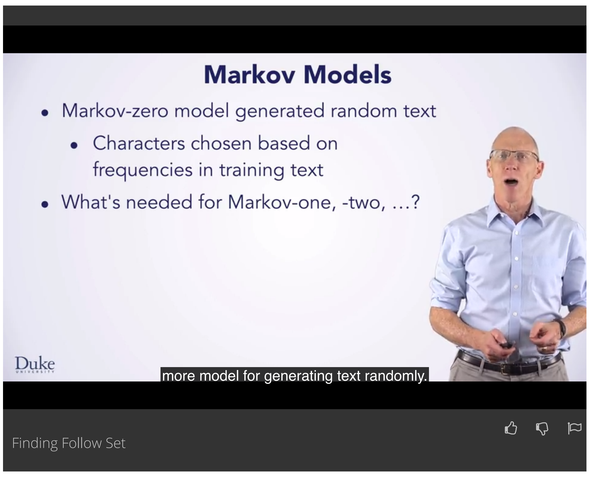

Finding Follow Set

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

Interactive Transcript

Search Transcript English

0:03
Hi, we'll use our seven step process to develop the key algorithm for Markov Models beyond Markov-zero, we've developed Java programs for the Markov-zero models of text generation. Characters were chosen at random from the training text, so if E occurs more often than A, it's more likely to be chosen in generating random text. After seeing the code to generate random text using a Markov-zero model, it's time to develop the key algorithm for implementing a Markov-one, two, or more model for generating text randomly, In a Markov-one model, if we generate t, we need to choose next from all characters that follow t. In a Markov-two model, if we generated th, we'd find all characters that follow every occurrence of th.

0:50
We use these follow characters to continue the process of random text generation. The key algorithm is finding these follow characters. We'll use our seven step process for a Markov-one model, but generalizing to two or more is very simple.

1:07
The first step in our seven step process is to work an instance. We'll use this sample training text that reads, this is an attempt, to illustrate the algorithm. We'll find the following characters of the letter t to develop the algorithm, but we'll generalize to any letter, not just to t, just as we'll generalize to any training text. The character h at index 1 follows the first t found at index 0. The next t is found at index 12. The following character is the t at index 13. Then comes the letter e, following the t found at index 13. We'll need to decide what to do with the t found at index 17. There are no follow characters, so we'll need to ignore this t, or choose another course of action when we develop the algorithm and write code.

1:59
Step two of a seven step process is to write down what we did. We start with a follow list that's empty, and we'll write down what we do to find characters that follow t in our training text. We search for the first occurrence of t starting at zero. We found the t at index zero, so we add the following character, an h at index 1 to follow.

2:24
Then we find a first t, starting at index 1, the first index we haven't examined in searching for a t.

2:32
Then we find the t at index 12, and add the t at index 13 to the follows list. We're ready to continue writing steps down, since we haven't searched all the training text.

2:44
Let's look at we get in writing down all the steps. Here are all the steps in getting to the list of letters that follow the key letter t in this training text. The follows list we have shows the three letters, but it's initially empty. We searched for a t starting at index zero and we found a t at index zero. We added the next character, the h at index one to the follow list, then we searched for a t starting at index one since that index hasn't been searched for. Not coincidentally, our search starts at the index of the most recent follow character. We find the t at index 12 and add the next character, the t at index 13, to the follow list. Then we search for a t starting at index 13, we find the t at that index, and so we add the next character, the e at index 14, to the follows list. Then we search for t starting at index 14. We find the t at index 17, but we can't add a follow character since there isn't one. So we stop and we have the complete follows list with three characters h, t, and e.

3:54
Now that we've written down all the steps we move on to step three, finding patterns and generalizing. We can see the process of finding an occurrence and adding the letter that follows in steps two and three of those steps we put down. That same pattern occurs at steps four and five for the next occurrence of t. Then the pattern occurs again at step six and seven. We've shown step eight as part of the same pattern, but there's no follow character so the pattern isn't completely duplicated.

4:27
In looking for patterns and generalizing we often try to connect the patterns we find. Steps two and three were the first occurrence of t. Steps four and five were the second occurrence, and steps six and seven were the last occurrence of t that had a follow character. Step eight was part of the same pattern, but it did not have a follow-up character. Note that the index of the follow character in one step Is the index at which the search starts in the following step. We find patterns and relations between them in generalizing what we wrote down.

5:05
As we get ready to write down the algorithm to test it with new data, we'll think about the steps we've written down. We'll think about how to initialize the algorithm and how to know when to stop our repeated steps. We start searching the training text at index 0 with an empty follows list. When will we know if we're done searching, so that we're done searching for key t in our example? If we don't find a t, say, after the last occurrence, we'll stop.

5:33
If we run off of the end of the training list, as with the last t, we'll also need to stop.

5:39
Now you're ready to write down the general algorithm. You'll start searching in index 0 with an empty follows list. We'll use the variable pos to indicate where we start the search.

5:50
While there's more searching to do, meaning we might find the key we're searching for as we loop through the steps, we find the first occurrence of KEY starting at pos. Remember that pos is initially zero but it changes in our loop. We store the location of the occurrence in variable index. If we find the key, we add it to the following list. We've used index as the variable of where the key is found, so index+1 is the next character since we're using a one character key. We update pos to be the index of the follow character. If we didn't find an occurrence of KEY in the previous step, we'd stop or break out of the loop.

6:38
Before you turn this into code, it's time to test out the algorithm. Give it a try on this training set, with the KEY being the letter e, excuse me, a.

6:51
Did you get the right answer?

6:55
If you're doing the algorithm carefully, you'd notice that even though you computed follows correctly, you never actually said follows is the answer. Oops, that's an easy mistake to make. But when you translate your algorithm into code, you have to explicitly return the answer you want. We better add that.

7:15
Now the algorithm produced the right answer, n, p, space and y. All follow the letter a. Happy coding.

Downloads

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

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