Text Generation in Python

AGENDA OF THE VIDEO

- What is text generation?
- How does it work?
- What can be done using text generator bots?
- Text generation in News.
- Methods of text generation (Two different approaches)
- Markov Chains.
- RNN + LSTM approach.
- How Markov chains can be used for text generation.
- Difference between markov method and RNN + LSTM approach.
- Implementing the Markov Chain function (A pseudocode approach).
- Implementing our text generator function.
- Task for you.

What is text generation?

Text generation as the name suggests is a method by which we give machine or computers the power to generate text.

The idea behind text generator bots is to have machine do our writing tasks. Also, in speech systems, these techniques are used by computers to hold conversations.

But how does that work?

How does that work?

If we have decided to generate some text, then we would first need to know beforehand that what is the style of text that we want to generate.

This step is very important. Because once we know the style, we would need the text corpus for the corresponding style.

Say, I want my machine to rap like eminem, then I would need lots of text in the form of eminem's songs lyrics.

So, text corpus is just a collection of text around a theme. Once, we have that, the goal of the text generation is to get some piece of text around that theme

What can be done?

If we can successfully build real text generators, then we can do lots of tasks like...

1. Book Writing
2. Email Writing
3. Song lyrics writer
4. Movie Writer
5. Etc etc

Text generation in news

As i am teaching this topics, text generation is in news.

The news is regarding the popular text generation technology that OpenAI has and how it is now planning to sell its technology.

There are many other news around this piece of technology regarding the potential danger of this technology because of improper use.

Methods of Text generation.

1. Markov Chain Methods,
2. Neural Network (Deep Learning Methods)

Markov Chain

What is a Markov Chain?

Markov chain is a mathematical concept related to sequence data. It says that for sequence the next state will only depend on its immediate previous state.

In a language, this means that the next word will only depend on its previous works which is somewhat overly simplified.

RNN +LSTM

Now, there is one more way by which one can generate text and that is by using a special neural network architecture and that is called "Recurrent Neural Network" architecture.

The only issue with them is that they are too slow to train. But once the training has been completed, the results will be surprisingly good.

Markov Chains for text generation

Given a text corpus. We convert it into tokens of words and then..

1. Compute a word to list of words dictionary.
2. Use that dictionary to generate new sequences.

Then what is the difference..

Difference is in terms of..
1. Speed
2. Performance
3. The way they work
4. Whether training is required or not
5. etc

Implement Markov Chain fun

```
## Markov Chain Function
## Input a text and return a markov chain of words and list of words

## Input text = text

1. get_tokens = text.split()

2. Intiatlize an empty dictionary

3. Loop through a zipped sequance of tokens

4. If the word is in the dictionary, then append the new word

5. Otherwise, make a new list

6. After the loop ends, return the dictionay
```

Implement our text generator

Given the markov chain of a give piece of text and the number of words to include in the new text, we can generate a new piece of text very easily by using the random module of python for selecting the new word to be included in the final generated text. Naturally, a word which occurs more will have more chances of having being included in the final text.

Task for you

Modify the generator function by

1. Adding some strict requirements for selecting a new word.
2. How should a punctuation should be included in the function?

Run the function on some other singer's lyrics and see what can you get. Does it make any sense?

Thank you