Markov Models for Language

STAT610 Final Project

Jiongran Wang
Yuanyuan Luan

11/23/2019

Abstract

In this model, language is assumed to be a sequence in which that next word is drawn from a distribution that depends on only the current word. We used the book named 'A Tale of Two Cities' from homework one as the 'pool' of words to estimate this distribution. Once the model has been built, we can gnerate a sequence of words with one single word as the 'starter' and one number that limiting the length of the sequence. In order to make sure the model is right, we have also written a test file that contians some relatively simple and short sentences which we can easily figure out the distribution. Therefore, we can double check it with the result of the model we built.

Github link

Data

We used the first book from the howework 1. first of all, we substracted the contents from the book and then split into single words, which looks like:

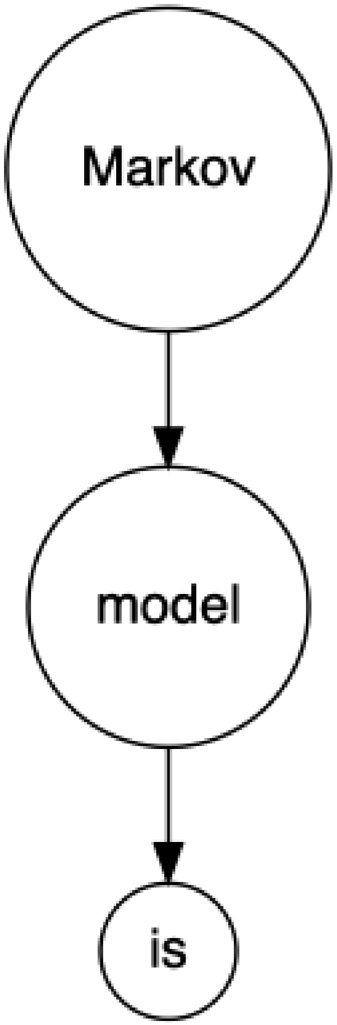
```
[1] "I."      "The"     "Period" "It"      "was"     "the"
```

Then we deleted all non-characters words from the data and capitalized all the words. After that, we find all the roman numbers and delete them. As the end, we delete all the empty strings that were created during the data cleaning process. We ended up with over $9,000$ words look like:

```
[1] "THE"     "PERIOD" "IT"      "WAS"     "THE"     "BEST"
```

Model

Given the property of Markov chains that the probabiliy of future states depends only on the cunrrent states, but not the sequence of previous states.The `Markov Models for Language` we built here is based on this property, which means what the next word is only depends on the what the present words is and so on.



For example, in the sentence “Markov model is a bit tedious but it is also a great model”, `Markov` comes with `model` right after it which is then followed by the word `is` and so on.

To buid this Markov chain with the first book for homework one, we first declared an empty list to store the result. Then, we iterate over all the words. We used all the unique words as the names of vectors in the list and every word that comes next to the unique words as the elements of the vector. For example, word `I` as the unique word, every word that comes next to every `I` will be the elements of the vector named `I`.

The first 6 elements in the vector for the firt unique word in our model looks like:

```
[1] "PERIOD" "BEST"   "WORST"  "AGE"    "AGE"    "EPOCH"
```

The name of this vector in the list is

```
[1] "THE"
```

This means in the vector named “THE”, we have all the words that come after “THE” in the book named “A Tale of Two Cities”.

Output

Once we have the distribution of all the words, we can generate a sequence of words given a start word and the length of the sequence. Here we used the “I” as the start word and limited the number of words in the sequence to be 10. Here are the sequence of words generated:

```
[1] "I MAKE MR LORRY NO A HUNDRED AND MISS HIS TRADE"
```