MOHAMMED V UNIVERSITY IN RABAT

National School of Computer Science and Systems Analysis

Research Master in Bioinformatic and Complex Systems Modeling Applied to Healthcare

Markov chain for Text Analysis

A Report

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Fatima Zohra MOADINE

06_26_2022

Supervised By: Professor, NAOUM Mohamed

Department of Computer Science and Decision Support,

ENSIAS

Contents:

List of abbreviation

MC– Markov Chain

NLP– Natural Language Processing

1. Introduction

Markov chain (MC) is a stochastic process(random), meaning it's based on random probability distribution. MC model is memory-less verifying the markov property_the prediction of the future state depends only on the present state. In addition, MC is broadly applicable in many real-world areas such as text generation[1],clinical NLP[2], genetics[3].
MC is applied for text modeling, applying different orders, i.e, tokenization of order one and three, character sequences or tokens of length one or three. Both token frequencies and probabilities of transition matrix are calculated to construct our model, and then testing it using another corpora.
First we'll define the steps for building our model, second we'll present the finding of our testing phase.

2. Methods and Results:

2.1. Data collection and Preprocessing

To build our Markov model, we'll first collect our data using python toolkits such as bs4[4], BeautifulSoup4[5], urllib.request[6], html.parser[7]. In addition, regular expressions are a very important step to remove unwanted parts of text, and build a specific structured text for our model. In this Lab, MC is used for predicting the next letter in our corpus, this technique was first used by Andrey Markov.

The training corpus of this Lab is the page web of the key word "Rabat" from Wikipedia, the testing corpus was of keyword "Casablanca".

2.2. Markov Chain Model

Coming next is splitting text by characters and white space, each character is a state of our model, while we are trying to predict the next character based on the previous one.
After cleaning of the text and tokenization, we move on to calculate the probability distribution of predicting the state, to build our transition matrix. The first step consists of both assigning an index to each letter of the alphabet, and second building a matrix p of 27 by 27, where each line and each column represents a state. The entry M[i,j] will be used to keep track of the number of times that the ith character of the alphabet is followed by the jth character of the alphabet.

The next step is to count all transitions using the following formula: For car from “1” to “the length of the first text - 1”, let i be the ith character in the text and j the (i+1)st character in the text. Then increment p[i,j].

Now, that all counts of all transitions are available, let's create the transition matrix using the following formula: For each k from 1 to 27, sum the entries on the ith row, i.e., let counter[k] = M[k,1] + M[k,2] + M[k,3] + ... + M[k,26], now define p[k, :] /= sum(p[k, :]) for all pairs i,j. This just gives a matrix of probabilities. In other words, now p[i,j] is the probability of making a transition from letter i to letter j.

The final step is testing the MC model on the second text. As a test the following might be implemented[8]:

- Keep track of a score, namely, initialize the "score" to 0.
- Keep track of the number of "attempts", namely, the number of times that we try to predict the next letter. Initialize the total number of attempts to 0.
- For each i from "1" to "the length of the second text - 1", let x be the ith character in the text and y be the (i+1)st character in the text.
- In the first half of the text, x is followed by y with probability p[x,y]. Let's consider this as a score, so increment the score by p[x,y], i.e., set score = score + P[x,y]. Also increment the number of attempts, i.e., set attempts = attempts + 1.
- Your overall fraction of correct attempts is score/attempts.

The fraction obtained training our model is equal to 0.112. Thus, the first text tell us so little about the second text.

3. Conclusion

The Markov assumption limits the history of each state to a fixed number of previous states[9]. This limited history is called the Markov order. In practice, using a higher -order Markov chain gives better results.

References

[1] D. Purwitasari, A. A. Zaqiyah, and C. Fatichah, “Word-Embedding Model for Evaluating Text Generation of Imbalanced Spam Reviews,” in *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2021, pp. 1–6. doi: 10.1109/ICACSIS53237.2021.9631315.

[2] J. Xu and T. Li, “Application of Multimodal NLP Instruction Combined with Speech Recognition in Oral English Practice,” *Mob. Inf. Syst.*, vol. 2022, p. e2262696, Apr. 2022, doi: 10.1155/2022/2262696.

[3] D. Almorza, A. Prada, M. V. Kandús, and J. C. Salerno, “An example to teach Markov chains and Hardy–Weinberg equilibrium through Mendel’s laws,” *J. Biol. Educ.*, vol. 0, no. 0, pp. 1–4, Apr. 2021, doi: 10.1080/00219266.2021.1905688.

[4] L. Richardson, *bs4: Dummy package for Beautiful Soup*. Accessed: Jun. 26, 2022. [Online]. Available: https://pypi.python.org/pypi/beautifulsoup4

[5] “Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation.” https://beautiful-soup-4.readthedocs.io/en/latest/ (accessed Jun. 26, 2022).

[6] “urllib.request — Extensible library for opening URLs — Python 3.10.5 documentation.” https://docs.python.org/3/library/urllib.request.html (accessed Jun. 26, 2022).

[7] “html.parser — Simple HTML and XHTML parser — Python 3.10.5 documentation.” https://docs.python.org/3/library/html.parser.html (accessed Jun. 26, 2022).

[8] “Markov Models for Text Analysis.” https://www.stat.purdue.edu/~mdw/CSOI/MarkovLab.html (accessed Jun. 26, 2022).

[9] D. Hovy, “Text Analysis in Python for Social Scientists: Prediction and Classification,” *Elem. Quant. Comput. Methods Soc. Sci.*, Feb. 2022, doi: 10.1017/9781108960885.