NEW MEXICO TECH
SCIENCE • ENGINEERING • RESEARCH UNIVERSITY

Class Project: Next word generation using Markov chain

Course Title (MATH-486-01-Intro to Stochastic Processes)

Md Ismail Hossain

May 5, 2022

Contents

List of Figures

List of Tables

1 Introduction

Natural Language Generation (NLG) is not a very new era of research. There are several way to generate words or even sentences. The use of Natural language processing and generation is tremendously increased last few years. We observed it's application in our daily life, as an example we all are using Siri, Google Assistant, Alexa in our day to day works. So, generation of text by machine or software is truly important. In this work I have used a very simple but powerful algorithm or logic of Markov chain in generating the next word or words.

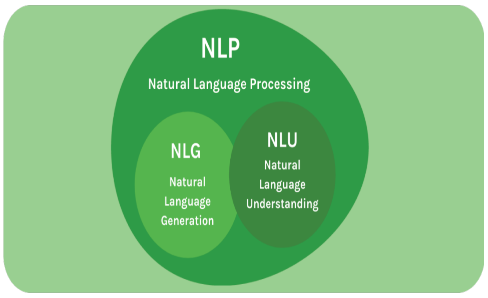
NLP
Natural Language Processing
NLG
Natural Language Generation
NLU
Natural Language Understanding

Figure 1: Language analytics domain.

2 Objective and Data

- Would calculate the Entropy of the train data to measure it's predictability.[2]
- The main objective is to use Markov chain to generate next words.[1, 3]

Used three different author works and applied the methodologies to predict the next word by Markov chain:

- Shakespeare [Macbeth] Data Source
- J.K. Rowling [The Philosopher's Stone] Data Source
- George R. R. Martin [Game of Thrones] Data Source

In this project I have used Python in analyzing the data. So, what I did:

Steps:

1. Calculate the Entropy of the entire document. (word level)
2. Construct the transition probability matrix (TPM).
3. Predict the next word or words using the TPM.

3 Methodology

Before feed the data in the Entropy calculation or transition probability matrix creation I have used few text processing steps for initial data cleaning like: tokenization and word mapping. Tokenization is to split the whole document in word level. I did not remove any punctuation because they also equally important as the words. Word mapping is necessary because software don't understand the texts, it only deal with the number. So, word mapping is also necessary when we are going to generate the next word.

Entropy

Entropy inherent the average uncertainty to the variables possible outcomes. It can be expressed using the function:

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_b P(x_i)$$

where,
X = Discrete random variables with possible outcome $x_1, x_2, x_3, ..., x_n$
$P(x_i)$ = The probability of occurring the i-th character.
b = The base of the logarithmic function.
For our case we would use b = length of the $x_1, x_2, x_3, ..., x_n$ to get the $H(X)$ value within 0 to 1.

Markov Property

In our case, x_t = process are in t-th character or word. So, using the Markov property, the x_{t+1}-th word or character only depends on x_t-th, not the previous words!

Transition Probability Matrix

By using the counts of the individual words or characters we can create the transition probability from i-th to j-th state:

$$p_{ij} = (X_{t+1} = j | X_t = i)$$

So, the transition probability matrix can be expressed as:

$$P = (p_{ij})$$

4 Results

In this section we are going to present the the Entropy calculation and the few predicted texts after generating the Transition probability matrix.

4.1 Entropy results

Table 1: Entropy values within 0 to 1

Book	**No. of unique words/charecter**	**Entropy**
Macbeth	16,874	0.71
The Philosopher's Stone	83,189	0.65
Game of Thrones	304,740	0.57

The Entropy value 1 indicating the perfect unpredictability or perfect randomness of the system here and 0 representing that only one possible outcome or word is presented in the system, so the generation of that word in next is certain. If there is a very large number of unique words in the document then the output of the process will be much more interesting.

Another interesting fact we can see from Table 1, when the number of unique words are increasing the entropy also decreasing. This might not be an unique findings but I could try more documents to verify whether there is any statistical relationship between unique number of words and the calculated entropy of the system.

4.2 Few Predicted texts

The key character name have used as the input text and then observed the output words from the Markov chain process. As Macbeth, Harry Potter and Jon Snow are the main or key character of the corresponding books

Book: Macbeth

Next 5 words after word Macbeth
'Macbeth Macb A foolish thought honest'

Next 10 words after word Macbeth
'Macbeth Rosse Ile be commanded heeres a TraitorAnd must be all'

Next 20 words after word Macbeth
'Macbeth MacbethBeware MacduffeBeware the Charme the Liars and pristine HealthI would be acted ere they comming on the heat of that'

Book: The Philosopher's Stone

Next 5 words after Harry
'Harry nodded vigorously. "Stop!'

Next 10 words after Harry
'Harry? I have a grin.. Professor Dumbledore gently'

Next 20 words after Harry
'Harry found an excellent adventure was another word on it had just one another point beating wouldn't be the board,'

Book: Game of Thrones

Next 5 words after Jon
'Jon Snow took a bloody cloak.'

Next 10 words after Jon
'Jon Snow with a blanket of snow crunched beneath his feet.'

Next 20 words after Jon
"Jon Snow, why is the King's Justice, you had a mind needs books as a whisper, and so it"

5 Conclusion

There are lots of complicated Machine learning algorithms, as an example Long short-term memory (LSTM) to generate next words which are much more accurate but in terms of computational complexity and training costs Markov Chain is one of the best choice because it's very easy to implement. But definitely for practical use on generating texts like creating a chat-bot, one must try more than one model to verify the performance of the algorithms. Moreover if we can train the model with a reasonable or large size of unique words, then the performance of next word generation would be much more logical and meaningful at least for our Markov chain model.

References

[1] Ashwin M J. Next Word Prediction using Markov Model. https://medium.com/ymedialabs-innovation/next-word-prediction-using-markov-model-570fc0475f96, 2018. [Online; accessed 16-March-2018].

[2] C. E. Shannon. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30(1):50–64, 1951.

[3] Luciano Strika. Markov Chains: How to Train Text Generation to Write Like George R. R. Martin. https://www.kdnuggets.com/2019/11/markov-chains-train-text-generation.html, 2019. [Online; accessed 29-November-2019].