

# Generating quote for William Shakespeare and Jane Austen Using Markov Text Generator

**Muktadir Chowdhury**

**U00534523**

## Introduction

Markov text generator is a basic form of text generator in Natural Language Generation (NLG) [1] which is a subfield of Natural Language Processing (NLP) [2]. An NLG system functions like a translator in that it translates data into a natural language representation. The conceptual workflow of NLG is reversed of that of Natural Language Understanding (NLU). NLU takes a natural language as an input and outputs machine representation of the language. On the other hand, NLG produces natural language from machine representation of a language. A simple example of an NLU system is the one that generates letter from a template.

## Related Work and Approach

In this work I used Markov model [4] to generate quotes of William Shakespeare and Jane Austen. Markov model is based on a stochastic theory called Markov chain and process introduced by Andrey Markov, a Russian Mathematician. Markov chain consists of states, where it is possible to make prediction about the next state by only looking at the current state. Since in this process we don't have to have knowledge about the previous states, the property of markov process is characterized as memoryless. The future states are independent of previous states and only dependent on the current states. A simple example of markov process can be part of my daily routine. Most days weekdays till evening I have three states: sleep, eat breakfast, go to lab. Now we can draw a state transition diagram from these states. Since one can predict my future state only by looking at my present state, it is a markov process.

We can use the same idea to generate text, if we can model text generation as a Markov process. In this specific process, each word will be a state, so we will predict word might occur after this word. If we list various words in a document and keep track of which words are coming after them, we can calculate the probability of transitioning from one word to another. We will also be able to calculate the probability and frequency of each word.

There are different stages of NLG [1]: content determination, document structuring, aggregation, lexical choice, referring expression generation and realization. In the content determination stage we need to decide what information to mention in the text. In this work we are focusing on the word usage of William Shakespeare and Jane Austen. The second and third stage, which are relevant to document generation, are not applicable to this project that only generates a single

sentence (quote). The fourth and fifth stage are also out of scope of this project. We implemented the sixth stage, realization, but we are not considering syntax and orthography.

## Implementation

I will use the following work of Jane Austen: Emma, Persuasion, Sense, and the following work of William Shakespeare: Caesar, Hamlet, and Macbeth to train our language model. Our text generation engine is built on top of the model. I have built two different version of the model.

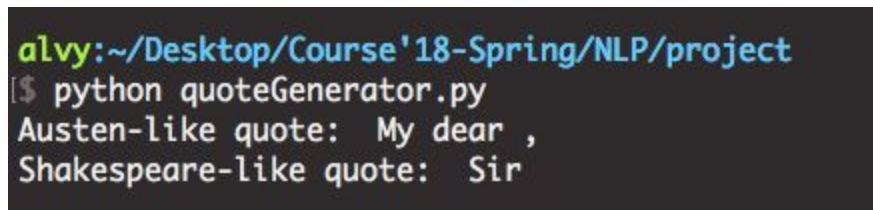
In the first, simplified version I build a bigram model, it contains which one word is preceding a word. The model contains a word and a list of words that are followed by that particular word. I have used two keywords, “START” and “END”, to store the words that are usually in the starting and ending of a sentence respectively. After building the model, we generate a quote by starting a word selecting from the “START” tag and keep picking words randomly until we encounter a word with the “END” tag. As we will see in the experiment and result section, the quote generated from this model is very short.

In the second model we built trigram model, i.e. when predicting a word we look back previous two words. With the trigram model the generated quotes are much better-looking. If we look back more words to predict next word, then we will have more better looking quotes. It is tempting to make our context sequence longer, but the longer we make them the more they are prone to overfitting the data and there will be no randomness. So what we did was to use the longest possible sequences so long as the pool of words following them is large enough. We are dynamically changing the length of our context sequence. If the pool of the next possible words are greater than some specified value then we choose one word from the pool. Otherwise, if the pool of the next words are smaller than the specified value, the generator fall back to a shorter context sequence.

Python is used as a programming language; we used Natural Language Toolkit (nltk) library [5].

## Experiments and Results

Below is the quote generated from the simplified model. As we can see that the quote is very short. It is because whenever the generator is encountering a word with “END”, it finishes the sentence.

A terminal window with a dark background and light-colored text. The prompt is 'alvy:~/Desktop/Course'18-Spring/NLP/project'. The command 'python quoteGenerator.py' has been executed. The output shows 'Austen-like quote: My dear ,' followed by a new line and 'Shakespeare-like quote: Sir'.

```
alvy:~/Desktop/Course'18-Spring/NLP/project
$ python quoteGenerator.py
Austen-like quote: My dear ,
Shakespeare-like quote: Sir
```

Below is the frequency of the top ten most frequent bigrams.

```

List of the top ten most frequent bigrams for Jane Austen:
(',',) : 28779
('to',) : 12021
('the',) : 11825
('and',) : 10761
('of',) : 10408
('a',) : 6576
('I',) : 6302
('her',) : 5976
('.',) : 5755
('was',) : 5561
List of the top ten most frequent bigrams for Shakespeare:
(',',) : 7052
('the',) : 1893
('"',) : 1750
(':',) : 1539
('I',) : 1412
('and',) : 1391
('to',) : 1257
('of',) : 1226
('you',) : 969
('a',) : 903

```

Below is the output of the program that generated quote using the back-off generation method.

```

alvy:~/Desktop/Course'18-Spring/NLP/project
$ python markovBackoff.py
Austen-like quote using markov backoff: enough for me to do such a knowledge of their numerous for intimacy with Mrs. Weston and their effect ; and, she reso
lved to
Shakespeare-like quote using markov backoff: . Hamlet give you goodnight, And heere from the Bondage you so slander any spurre, yet I have that Within a Fren
ch, and
alvy:~/Desktop/Course'18-Spring/NLP/project
$

```

**Austen-like quote using markov backoff:** enough for me to do such a knowledge of their numerous for intimacy with Mrs. Weston and their effect ; and, she resolved to

**Shakespeare-like quote using markov backoff:** . Hamlet giue you goodnight, And heere from the Bondage you so slander any spurre, yet I haue that Within a French, and

As we can see from the output the quotes are not complete nonsensical and bear resemblance to the respective authors.

## Conclusions and Future Work

Due the shortage of time I could not evaluate my model. The model can be evaluated by calculating the probability of the quote it generates and finding out how similar it is to original authors' quote.

## References

- [1] Bateman, J., & Zock, M. (2003). Natural language generation. In *The Oxford Handbook of Computational Linguistics* 2nd edition.
- [2] Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1), 51-89.
- [3] Allen, J. (1995). *Natural language understanding*. Pearson.
- [4] Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *ieee assp magazine*, 3(1), 4-16.
- [5] <http://www.nltk.org/>