import random

import nltk

from nltk.tokenize import word\_tokenize

# Download required data (run only once)

nltk.download('punkt')

class MarkovChainStory:

def \_init\_(self, text, n=2): # <-- constructor with arguments

self.n = n

self.model = {}

self.build\_model(text)

def build\_model(self, text):

tokens = word\_tokenize(text.lower())

for i in range(len(tokens) - self.n):

prefix = tuple(tokens[i:i+self.n])

next\_word = tokens[i+self.n]

if prefix not in self.model:

self.model[prefix] = []

self.model[prefix].append(next\_word)

def generate(self, size=50):

start = random.choice(list(self.model.keys()))

output = list(start)

for \_ in range(size):

prefix = tuple(output[-self.n:])

if prefix in self.model:

next\_word = random.choice(self.model[prefix])

output.append(next\_word)

else:

break

return " ".join(output).capitalize() + "."

# Example training text

training\_text = """

Once upon a time in a faraway land, there lived a brave knight.

The knight fought dragons and protected the villagers.

One day, he discovered a hidden cave with treasure.

But inside the cave, a sleeping dragon was waiting.

"""

# Build and generate story

story\_gen = MarkovChainStory(training\_text, n=2) # ✅ now works

print(story\_gen.generate(80))s