Task-03: Text Generation with Markov Chains

## Introduction

Markov Chains are probabilistic models that predict the next state based on the current state. In this task, we implement a simple text generation model that uses Markov Chains to predict the next word or character based on the previous one.

## Steps Involved

1. Read and preprocess the input text  
2. Build a Markov Chain model (dictionary of word transitions)  
3. Generate new text by randomly choosing next words based on probabilities

## Sample Code

```python  
import random  
  
def build\_markov\_chain(text, n=1):  
 words = text.split()  
 markov\_chain = {}  
 for i in range(len(words) - n):  
 key = tuple(words[i:i+n])  
 next\_word = words[i+n]  
 if key not in markov\_chain:  
 markov\_chain[key] = []  
 markov\_chain[key].append(next\_word)  
 return markov\_chain  
  
def generate\_text(chain, n=1, length=50):  
 start = random.choice(list(chain.keys()))  
 result = list(start)  
 for \_ in range(length):  
 key = tuple(result[-n:])  
 next\_words = chain.get(key)  
 if not next\_words:  
 break  
 result.append(random.choice(next\_words))  
 return ' '.join(result)  
  
# Example usage  
with open("sample\_text.txt") as f:  
 text = f.read()  
chain = build\_markov\_chain(text)  
print(generate\_text(chain))  
```

## Example Output

Generated text: 'The sun was shining bright and the birds were singing in the trees as he walked along the path.'

## How to Run

1. Save your input text in 'sample\_text.txt'  
2. Run the script in Python  
3. The output will be a generated sentence based on the Markov model

## References

1. https://en.wikipedia.org/wiki/Markov\_chain  
2. https://towardsdatascience.com/text-generation-using-markov-chains-515c1e242cd3