Implement a simple text generation algorithm using Markov chains in Python.

```python
import random

def build_markov_chain(text, n=1):
    """
    Build a Markov chain model.

    :param text: Input text (string)
    :param n: Order of the Markov chain (number of previous words to
consider)
    :return: Dictionary representing the Markov chain
    """
    words = text.split()
    markov_chain = {}

    for i in range(len(words) - n):
        # Create the state from previous n words
        state = tuple(words[i:i+n])
        next_word = words[i+n]

        if state not in markov_chain:
            markov_chain[state] = []
        markov_chain[state].append(next_word)

    return markov_chain

def generate_text(chain, length=50, n=1):
    """
    Generate text using the Markov chain model.

    :param chain: Markov chain dictionary
    :param length: Number of words to generate
    :param n: Order of the Markov chain
    :return: Generated text string
    """
    state = random.choice(list(chain.keys()))
    output = list(state)

    for _ in range(length - n):
        next_words = chain.get(state)
        if not next_words:
            break
        next_word = random.choice(next_words)
        output.append(next_word)
        state = tuple(output[-n:])

    return ' '.join(output)

# Example usage:
```

```
sample_text = """
Markov chains are a simple yet powerful method to model sequences.
They can predict the probability of the next item in a sequence
based on the previous ones. This can be used to generate text
that mimics the style of the original input.
"""

# Build a Markov chain with 2-word context
chain = build_markov_chain(sample_text, n=2)
generated_text = generate_text(chain, length=30, n=2)
print(generated_text)
```