Here we will show the model performance against the test case

In [ ]:

```python
"""Markov Text Generator.

Patrick Wang, 2023

Resources:
Jelinek 1985 "Markov Source Modeling of Text Generation"
"""

import nltk

from mtg import finish_sentence
```

In [ ]:

```python
"""Test Markov text generator."""
corpus = nltk.word_tokenize(nltk.corpus.gutenberg.raw("austen-sense.txt").lower())

words = finish_sentence(
    ["she", "was", "not"],
    3,
    corpus,
    randomize=False,
)
print(words)
```

```
['she' 'was' 'not' 'in' 'the' 'world' '.']
```

In [ ]:

```python
words2 = finish_sentence(
    ["robot"],
    3,
    corpus,
    randomize=False,
)
print(words2)
```

```
['robot' ',' 'and' 'the' 'two' 'miss' 'steeles' ',' 'as' 'she']
```

In [ ]:

```python
words3 = finish_sentence(
    ["she", "was", "not"],
    1,
    corpus,
    randomize=False,
)
print(words3)
```

```
['she' 'was' 'not' ',' ',' ',' ',' ',' ',' ',']
```

In [ ]:

```python
words4 = finish_sentence(
    ["robot"],
    2,
    corpus,
    randomize=False,
)
print(words4)
```

```
['robot' ',' 'and' 'the' 'same' 'time' ',' 'and' 'the' 'same']
```

Next we will look at the random case. We will use a larger n because it produces more interesting sentences and see that for the same initial parameters, different sentences are selected: (we will start with a deterministic pass to see the output)

In [ ]:

```
words5_5 = finish_sentence(
    ["she", "was", "not"],
    4,
    corpus,
    randomize=False,
)
print(words5_5)
```

```
['she' 'was' 'not' 'in' 'the' 'house' ',' 'and' 'they' 'were']
```

Now the ouputs from randomly selected candidate words

In [ ]:

```
words5 = finish_sentence(
    ["she", "was", "not"],
    4,
    corpus,
    randomize=True,
)
print(words5)
```

```
['she' 'was' 'not' 'suspected' 'of' 'any' 'extraordinary' 'interest' 'in'
 'it']
```

In [ ]:

```
words6 = finish_sentence(
    ["she", "was", "not"],
    4,
    corpus,
    randomize=True,
)
print(words6)
```

```
['she' 'was' 'not' 'aware' 'that' 'such' 'language' 'could' 'be'
 'suffered']
```

The above is actually a completely legible sentence which is interesting

In [ ]:

```
words7 = finish_sentence(
    ["she", "was", "not"],
    4,
    corpus,
    randomize=True,
)
print(words7)
```

```
['she' 'was' 'not' 'doomed' ',' 'however' ',' 'to' 'be' 'sure']
```

That is the conclusion of this markov text generator