```
import math
import nltk
from nltk.util import ngrams
from nltk.lm import MLE
from nltk.lm.preprocessing import padded_everygram_pipeline
```

```
# Sample corpus
text = [
    ["this", "is", "a", "test"],
    ["this", "is", "another", "test"],
    ["we", "are", "testing", "language", "model"]
]

# Step 1: Preprocess data for bigram model
n = 2  # bigram
train_data, vocab = padded_everygram_pipeline(n, text)

# Step 2: Train the MLE model
model = MLE(n)
model.fit(train_data, vocab)

# Step 3: Test sentence
test_sentence = ["this", "is", "a", "test"]

# Step 4: Compute perplexity
# Prepare test data
test_data, _ = padded_everygram_pipeline(n, [test_sentence])

ppl = model.perplexity(list(next(test_data)))

print("Test Sentence:", " ".join(test_sentence))
print("Perplexity:", ppl)
```

```
Test Sentence: this is a test
Perplexity: 3.7327440761235615
```