

Hidden Markov Model

```
!pip install hmmlearn numpy
```

```
Requirement already satisfied: hmmlearn in /usr/local/lib/python3.12/dist-packages (0.3.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (2.0.2)
Requirement already satisfied: scikit-learn!=0.22.0,>=0.16 in /usr/local/lib/python3.12/dist-packages (from hmmlearn) (1.6.1)
Requirement already satisfied: scipy>=0.19 in /usr/local/lib/python3.12/dist-packages (from hmmlearn) (1.16.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn!=0.22.0,>=0.16->hmmle
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn!=0.22.0,>=0.16
```

```
import numpy as np
from hmmlearn import hmm
```

```
# Toy training sentences
sentences = [
    ["the", "dog", "barks"],
    ["a", "cat", "meows"],
    ["the", "cat", "runs"],
]

# Corresponding POS tags
# DET = Determiner, NOUN = Noun, VERB = Verb
tags = [
    ["DET", "NOUN", "VERB"],
    ["DET", "NOUN", "VERB"],
    ["DET", "NOUN", "VERB"],
]
```

```
# Unique words and tags
word_vocab = list({word for sent in sentences for word in sent})
tag_vocab = list({tag for sent in tags for tag in sent})

# Maps
word_to_id = {w: i for i, w in enumerate(word_vocab)}
tag_to_id = {t: i for i, t in enumerate(tag_vocab)}

id_to_tag = {i: t for t, i in tag_to_id.items()}

# Print vocab for clarity
print("Word Vocabulary:", word_to_id)
print("Tag Vocabulary:", tag_to_id)
```

```
Word Vocabulary: {'the': 0, 'dog': 1, 'runs': 2, 'cat': 3, 'barks': 4, 'meows': 5, 'a': 6}
Tag Vocabulary: {'VERB': 0, 'DET': 1, 'NOUN': 2}
```

```
# Convert words and tags into numerical format
X = [[word_to_id[w] for w in sent] for sent in sentences]
Y = [[tag_to_id[t] for t in tag_list] for tag_list in tags]

# HMM requires 1D arrays + lengths
X_concat = np.concatenate([np.array(x) for x in X])
lengths = [len(x) for x in X]

print("X_concat:", X_concat)
print("lengths:", lengths)
```

```
X_concat: [0 1 4 6 3 5 0 3 2]
lengths: [3, 3, 3]
```

```
model = hmm.MultinomialHMM(n_components=len(tag_vocab), n_iter=50)

model.fit(X_concat.reshape(-1,1), lengths)

print("Training completed!")
```

WARNING:hmmlearn.hmm:MultinomialHMM has undergone major changes. The previous version was implementing a CategoricalHMM (a specification of a Hidden Markov Model)
<https://github.com/hmmlearn/hmmlearn/issues/335>
<https://github.com/hmmlearn/hmmlearn/issues/340>
Training completed!

```
test_sentence = ["the", "cat", "barks"]

# Convert words to ids
test_ids = np.array([word_to_id[w] for w in test_sentence]).reshape(-1,1)

# Predict hidden states (POS tags)
predicted_tag_ids = model.predict(test_ids)

# Convert back to text tags
predicted_tags = [id_to_tag[i] for i in predicted_tag_ids]

print("Sentence: ", test_sentence)
print("Predicted POS Tags:", predicted_tags)
```

```
Sentence:  ['the', 'cat', 'barks']
Predicted POS Tags: ['NOUN', 'VERB', 'DET']
```

Start coding or [generate](#) with AI.