```
# ================================================
# Лабораторна робота: POS-тегування на основі HMM
# Корпус: Sinica Treebank (Traditional Chinese)
# ================================================

import nltk
from nltk.corpus import sinica_treebank
from collections import defaultdict, Counter
import random

# ==============================
# 0. Завантаження корпусу
# ==============================
nltk.download('sinica_treebank', quiet=True)

tagged_sents = list(sinica_treebank.tagged_sents())
print(f"Кількість речень у корпусі: {len(tagged_sents)}")

# Для стабільності результатів
random.seed(42)
random.shuffle(tagged_sents)

# Розділення на train/test (80/20)
split = int(0.8 * len(tagged_sents))
train_sents = tagged_sents[:split]
test_sents = tagged_sents[split:]
print(f"Розмір тренувальної вибірки: {len(train_sents)}, тестової: {len(test_sents)}\n")

# ==============================
# 1. Побудова словників частот
# ==============================
transition_counts = defaultdict(Counter)
emission_counts = defaultdict(Counter)
tag_counts = Counter()

for sentence in train_sents:
    prev_tag = '<START>'
    for word, tag in sentence:
        transition_counts[prev_tag][tag] += 1
        emission_counts[tag][word] += 1
        tag_counts[tag] += 1
        prev_tag = tag
    transition_counts[prev_tag]['<END>'] += 1

# ==============================
# 2. Обчислення ймовірностей
# ==============================
# Матриця переходів A
A = defaultdict(dict)
for prev_tag, counter in transition_counts.items():
    total = sum(counter.values())
    for tag, count in counter.items():
        A[prev_tag][tag] = count / total

# Матриця емісій B
B = defaultdict(dict)
for tag, counter in emission_counts.items():
    total = sum(counter.values())
    for word, count in counter.items():
        B[tag][word] = count / total

all_tags = list(tag_counts.keys())

# ==============================
# 3. Алгоритм Вітербі
# ==============================
def viterbi(sentence, A, B, all_tags):
```

```
    V = [{}]
    path = {}

    # Початковий крок
    for tag in all_tags:
        trans_p = A['<START>'].get(tag, 1e-6)
        emis_p = B[tag].get(sentence[0], 1e-6)
        V[0][tag] = trans_p * emis_p
        path[tag] = [tag]

    # Основний цикл
    for t in range(1, len(sentence)):
        V.append({})
        new_path = {}

        for curr_tag in all_tags:
            best_prob, best_prev = 0, None
            emis_p = B[curr_tag].get(sentence[t], 1e-6)
            for prev_tag in all_tags:
                prev_prob = V[t-1].get(prev_tag, 0)
                trans_p = A[prev_tag].get(curr_tag, 1e-6)
                prob = prev_prob * trans_p * emis_p
                if prob > best_prob:
                    best_prob, best_prev = prob, prev_tag
            if best_prev is not None:
                V[t][curr_tag] = best_prob
                new_path[curr_tag] = path[best_prev] + [curr_tag]

        path = new_path

    # Завершальний крок
    n = len(sentence) - 1
    best_final_tag = max(V[n], key=V[n].get)
    return path[best_final_tag]

# ==============================
# 4. Оцінка точності
# ==============================
def evaluate(test_sents, A, B, all_tags):
    total, correct = 0, 0
    for sentence in test_sents:
        words = [w for w, _ in sentence]
        true_tags = [t for _, t in sentence]
        pred_tags = viterbi(words, A, B, all_tags)
        for p, t in zip(pred_tags, true_tags):
            if p == t:
                correct += 1
            total += 1
    return correct / total

accuracy = evaluate(test_sents, A, B, all_tags)
print(f"\nТочність HMM POS-тегера: {accuracy*100:.2f}%\n")

# ==============================
# 5. Приклади тегування
# ==============================
print("Приклади розмічення речень:")
for i, sent in enumerate(test_sents[:3]):
    words = [w for w, _ in sent]
    tags_pred = viterbi(words, A, B, all_tags)
    print(f"Речення {i+1}:")
    print("Слова: ", words)
    print("Теги:  ", tags_pred)
    print()

# ==============================
# 6. Порівняння з NLTK (англ. тегування)
# ==============================
from nltk import pos_tag
```

```
english_example = ['This', 'is', 'an', 'example', '.']
print("Приклад NLTK POS-тегера (англійська):")
print(pos_tag(english_example))
```

```
Кількість речень у корпусі: 10000
Розмір тренувальної вибірки: 8000, тестової: 2000
```