

```

from collections import Counter

def add_one_smoothing_bigrams(corpus):
    # Tokenize the corpus into bigrams
    bigrams = [tuple(corpus[i:i+2]) for i in range(len(corpus)-1)]

    # Count the occurrences of each bigram
    bigram_counts = Counter(bigrams)

    # Count the occurrences of each unigram (word)
    unigram_counts = Counter(corpus)

    # Calculate vocabulary size (unique words)
    vocabulary_size = len(set(corpus))

    # Apply Add-One (Laplace) smoothing to calculate bigram
    probabilities
    add_one_smoothed_probs = {}
    for bigram, count in bigram_counts.items():
        # Bigram smoothing formula:
        #  $P(w_2|w_1) = \frac{\text{Count}(w_1, w_2) + 1}{\text{Count}(w_1) + V}$ 
        w1, w2 = bigram
        bigram_prob = (count + 1) / (unigram_counts[w1] +
vocabulary_size)
        add_one_smoothed_probs[bigram] = bigram_prob

    return add_one_smoothed_probs

# Example corpus
corpus = ["the", "quick", "brown", "fox", "jumps", "over", "the",
"lazy", "dog"]

# Apply add-one smoothing to bigrams
add_one_smoothed_probabilities = add_one_smoothing_bigrams(corpus)

# Print the add-one smoothed probabilities
for bigram, prob in add_one_smoothed_probabilities.items():
    print(f"Bigram: {bigram}, Smoothed Probability: {prob:.4f}")

Bigram: ('the', 'quick'), Smoothed Probability: 0.2000
Bigram: ('quick', 'brown'), Smoothed Probability: 0.2222
Bigram: ('brown', 'fox'), Smoothed Probability: 0.2222
Bigram: ('fox', 'jumps'), Smoothed Probability: 0.2222
Bigram: ('jumps', 'over'), Smoothed Probability: 0.2222
Bigram: ('over', 'the'), Smoothed Probability: 0.2222
Bigram: ('the', 'lazy'), Smoothed Probability: 0.2000
Bigram: ('lazy', 'dog'), Smoothed Probability: 0.2222

```