# MP4

# 1 Forward Algorithm and Backward Algorithm

1. To compute using forward algorithm, we need

, , , ,

, ,

0.01967

Finally,

1. To compute using backward algorithm, we need

, , , , , , ,

Finally,

# 2 Viterbi Algorithm

1. Start probability with top 10 words with highest output probabilities:

'NNP': 0.5602442510068858, ('County', 'New', 'England', 'River', 'India', 'South', 'District’, ‘US', 'Crambidae', 'School')

'NNS': 0.011952708847602963, ('species', 'stars', 'roles', 'forests', 'moths', 'sports', 'areas', 'people', 'services' 'spiders')

'DT': 0.34473171365467065, ('the', 'a', 'The', 'an', 'This', 'A', 'this', 'both', 'all', 'An')

'IN': 0.017279459529686892, ('of', 'in', 'by', 'on', 'as', 'at', 'for', 'with', 'from', 'near')

'JJ': 0.019929842795894503, ('American', 'historic', 'small', 'Indian', 'former', 'geologic', 'civil', 'public', 'national', 'southern')

'VBN': 0.006548005716512927, ('located', 'known', 'based', 'owned', 'been', 'used', 'situated', 'named', 'operated', 'written')

'RB': 0.009640119527088476, ('also', 'currently', 'now', 'as', 'well', 'not', 'commonly', 'just', 'approximately', 'only')

'NN': 0.028738469533584513, ('family', 'moth', 'village', 'district', 'state', 'town', 'area', 'station', 'genus', 'film')

'CC': 0.0006236195920488502, ('and', 'or', 'but', 'either', 'both', 'moth', 'And', 'et', 'plus', 'But')

'VBZ': 0.0003118097960244251, ('is', 'has', 'consists', 'includes', 'serves', 'lies', 'features', 'contains', 'plays', 'provides')

1. If test sentence contains some word that is never seen in training data, the model trained in will fail as the output probability

for all tag sentences .

My solution: randomly assign a tag and a probability to the word which is seen in test data but never seen in training set.

1. Sentence-level accuracy for test\_0: 0.79

Word-level accuracy for test\_0: 0.9783

Sentence-level accuracy for test\_1: 0.07

Word-level accuracy for test\_1: 0.902