1 hidden Markov Model(HMM) for part-of-speech (POS)

tagging, using the Brown corpus as training data. The tag set is

composed of the twelve POS tags: Noun (noun), Verb (verb), Adj

(adjective), Adv (adverb), Pron (pronoun), Det (determiner or

article), Adp (preposition or postposition), Num (numeral), Conj

(conjunction), Prt (particle), ‘.’ (punctuation mark) and x (other).

a) A function ����\_������(���h) that loads the corpus at the

given path and returns it as a list of POS-tagged sentences. Each line

in the file should be treated as a separate sentence, where sentences

consist of sequences of white space-separated strings of the

form "����� = ���". Your function should return a list of lists, with

individual entries being 2-tuples of the form (token, POS).

b) Write an initialization method which takes a list of sentences in

the form produced by load\_corpus(path) as input and initializes the

internal variables needed for the POS tagger. In particular, if

�!,�!,�!,�!, ⋯ ,�! denotes the set of tags and �!, �!, �!, �!, ⋯ , �!

denotes the set of tokens found in the input sentences, computation of the

following:

• The initial tag probabilities �(�!) for 1 < � < �, where �(�!) is

the probability that a sentence begins with tag �!.

• The transition probabilities �(�! ⟶ �!) for 1 < �,� < � where

�(�! ⟶ �!) is the probability that tag �! occurs after tag �!.

• The emission probabilities �(�! ⟶ �!) for 1 < � < � and

1 < � < � where �(�! ⟶ �!) is the probability that token �! is

generated given tag �!

c) Write a method ����\_��������\_����(�����, ��� ���������) which

returns the list of the most probable tags corresponding to each input

word-token. In particular, the most probable tag for a token, wj is

defined to be the tag with index �

∗ = ������! �(�! →

�!). Computation will likely proceed in two stages: you will first

compute the probability of the most likely tag sequence, and will then

reconstruct the sequence which achieves that probability from end to

beginning by tracing backpointers.