4.1 For this question, I have printed frequency counts of all wordtag, unigram tags and bigram tags in below mentioned file names.

$C(w_i; t_i)$: wordtagCounts.txt
$C(t_i)$: unigramtagCounts.txt
$C(t_{i-1}; t_i)$: bigramtagCounts.txt

In wordtagCounts.txt, at each line we have a unique word tag pair with its count in format 'word/tag : count'
In unigramtagCounts.txt at each line we have a POS tag with its count in format 'tag : count'. For this question, we have added <s> and <e> as the start and end tag of each sentence in each file.
Exception : Even though In unigramtagCounts.txt contains POS tags , we have counts of <s> tag as well.

In bigramtagCounts.txt each line we have a POS tag pair with its count in format 'previoustag currenttag : count'

Exception : In this file , we may see <s> and <e> tags along with all other POS tags as these tags were needed for calculating transition probability

Note: I have not replaced low frequency words with 'UNK' in this question as I have handled unknown word in test data by doing smoothing

4.2 Transition Probability with smoothing
Output of this question is in transitionProb.txt file. Format of output is
' previousTag currentTag : probablity'
I have used smoothing by using lenth of bigramDict as the vocabulary size.

4.3 Emission Probablity with smoothing
Output of this question is in emissionProb.txt file. Format of output is
' currentWordTag currentTag : probablity'
I have used smoothing by using lenth of wordTagDict as the vocabulary size.

4.4 Output of this question is in the main note book named as q4_Viterbi.ipynb. I have pasted the same below as well. We can generate different sentences by running that particular cell again.

<s> I/ppss represents/vbz hall/nn-hl Need/nn-hl of/in-hl Norms/nns-hl meteorites/nns-hl in/in-hl in/in-hl an/at-hl A/at-hl The/at-hl the/at-hl the/at-hl dream/nn-hl warfare/nn-hl co-optation/nn-hl iodine/nn-hl off/rp-hl <e> 1.0812655701071951e-85

<s> Quint/np limitations/nns intervals/nns ,/, I/ppss you/ppss sent/vbd his/pp$ his/pp$ poems/nns effect/nn measurements/nns article/nn and/cc &/cc-tl Government/nn-tl Tumor/nn-tl Island/nn-tl President/nn-tl ./. <e> 1.070531057195176e-62

<s> 2/cd-hl <e> 3.853427882184323e-05

<s> a/at appreciated/vbd Santa/np Manchester/np-tl Chicago/np-tl Figure/nn-tl Morris/np-tl Sherman's/np$ expressivness/nn or/cc elementary/jj question/nn what/wdt-hl which/wdt He/pps Did/dod did/dod did/dod did/dod didn't/dod* did/dod didn't/dod* didn't/dod* written/vbn a/at mat/nn start/nn must/md a/at-hl An/at-hl the/at-hl method/nn-hl chemical/nn-hl water/nn-hl device/nn-hl Life/nn-hl of/in-hl of/in-hl in/in-hl of/in-hl the/at-hl portrayed/vbn-hl ./.-hl ?/.-hl <e> 4.029490713778099e-169

<s> when/wrb ?/. <e> 5.1373250926109715e-05

Observation: Sometimes the probability of long sentences comes as 0.0. This happens because we are multiplying transitionProb and emissionProb whose range is in 0 to 1 and with long sentences value goes out of range and becomes 0.

4.5 In this question, I am using 2 functions to generate POS Tag using Viterbi algorithm **findTagsPerSentenceUsingList function** is generating list of list between each word of a given sentence and each tag possible.

There were many unknown words in test dataset , I handled them by using smoothing on training data i.e. in case I encounter any unknown word, I calculate its basictransitionProb and basicemissionProb by taking it's transition and wordtag count as zero as the word is unseen and then select the best possible tab based on the probability score.

generateTaggedSentenceUsingList function is using df and bp values output from previous function and assigning the best possible tag to each word and returning a tagged sentence to output.

Output of this question is in viterbiOutput.txt file.