MIS: 111708049 Assignment Topic - Implement Viterbi Algorithm with backtracking
Files used : Hindi-tagged-[1-12, 14, 15].txt = 14 files
Preprocessing on corpus :

Remove references of type
Replace eclipse with space
Some tags had double underscore, remove the extra underscore
Separate commas “word,_tag” -> changed to “word_tag ,_RD_PUNC”
Add RD_PUNC tag for punctuations where its not present
Add RD_SYM tag for symbols where its not present
Start symbol - “<s>”, End Symbol - “<\s>”
Adding start and end symbol if not present

Handling sentence ends : Added _SENT tag for sentence ending with ["?", "!", ".", "।", "|"]

Separating word and tags :

IF ‘_’ present :Separate on ‘_’ to get word and tag associated,
ELSE :
IF word = “<s>”, tag = “START”
ELSE IF word = “<\s>”, tag = “END”
ELSE : tag = “UN”

Get_ngrams : returns ngrams using the list of 1-grams(words) provided
Get_freq_dict : returns frequency of each ngram in list ngrams

Tag_Transition_Matrix(dict) : get all tag bigrams from the list of tags in corpus(sequential),

IF bigram in corpus :
TTP = count(tag bigrams in corpus)/count(tag bigram[0])
ELSE :
TTP = 0.0001

Word Emission probability(dict) : Used add-k smoothing for handling unknown words in test data, |V| = no of distinct words in corpus, get all word-tag pairs,

IF word-tag pair in corpus :
WEP = count(word, tag)+k/count(tag)+k*V
ELSE :
WEP = k/count(tag)+k*V

Viterbi Algorithm : return tagged sentence, given test_sentence, words, tags in corpus(sequential)
Get WEPs, TTPs, SEQSCORE = np.zeros((T, N)), BAKCPTR = np.zeros((T, N))
Initialization Step
For i = 0 to N-1 do

//N is no of lexical categories and T is no of words
SEQSCORE (0,i) = P (W 1 |C i) * P (C i |”START”)

Iteration Step
For t = 1 to T-1 do
    For i = 0 to N-1
        SEQSCORE (t,i) = MAX j=1,N (SEQSCORE (t-1,j) * P(C i |C j)) * P(w t | Ci)
        BACKPTR(t,i) = index of j that gave max above
Sequence Identification Step
C(T-1) = i that maximizes SEQSCORE (T-1,i)
For i = T-2 to 0 do
    C(i) = BACKPTR(i+1,C(i+1)) //Back trace to find the sequence

tags[C[i]] gives tag for word test_tokens[i]

Preprocess test_sentence : Add “<s>”, “</s>” if not present

Main :
    Create hindi_tags_list, hindi_words_list from corpus s.t. Tag of hindi_words_list[i] if given at hindi_tags_list[i] (read sents, preprocess, handle sent ends)
    Make TTP, WEP files using respective functions
    Get tagged sentences, given unlabelled sentence, and write it to output file.

111708049_Assign3_Viterbi_Input.txt :
आज होगा जबरदस्त मुकाबला क्रिकेट का !
सरकार हमारी जरूरतों को पूरा करे ।
इस नीति के बारे में कुछ भी सही नहीं है ।
आपने क्या उस जगह पर जाने की कोशिश की थी जो मैंने आपको बताया था ?
कई दशक पहले यहां एक बड़ा बरगद का पेड़ हुआ करता था ।
उन्होंने कहा - भारत में आपका स्वागत है !
दिन के अंत तक हम कितनी दूरी तय कर सकते हैं ?
राजा और रानी बहुत घमंडी थे |
मेरे पास फिल्म को निर्देशित करने के लिए पैसे नहीं हैं |
नदी के पास एक झोपड़ी है इसलिए हम वहां डेरा डाल सकते हैं !

111708049_Assign3_Viterbi_Output.txt :
आज_N_NN होगा_V_VM जबरदस्त_JJ मुकाबला_N_NN क्रिकेट_N_NN का_PSP !_SENT
सरकार_N_NN हमारी_PRP जरूरतों_N_NN को_PSP पूरा_N_NN करे_V_VM ।_SENT
इस_DM_DMD नीति_N_NN के_PSP बारे_N_NN में_PSP कुछ_QT_QTF भी_RP_RPD
सही_N_NN नहीं_RP_NEG है_V_VAUX ।_SENT
आपने_PRP क्या_PRQ उस_PRP जगह_N_NN पर_PSP जाने_V_VM की_PSP कोशिश_N_NN
की_PSP थी_N_NN जो_PRP मैंने_PRP आपको_PRP बताया_V_VM था_V_VAUX ?_SENT
कई_QT_QTF दशक_N_NN पहले_IN यहां_N_NN एक_QT_QTC बड़ा_JJ बरगद_N_NN
का_PSP पेड़_N_NN हुआ_V_VM करता_V_VM था_V_VAUX ।_SENT
उन्होंने_N_NN कहा_V_VM -_RD_SYM भारत_N_NNP में_PSP आपका_PRP स्वागत_N_NN
है_V_VM !_SENT

दिन_N_NN के_PSP अंत_N_NN तक_PSP हम_PRP कितनी_QTF दूरी_N_NN तय_N_NN कर_V_VM सकते_V_VAUX हैं_V_VAUX ?_SENT
राजा_N_NN और_CCD रानी_N_NN बहुत_PSP घमंडी_V_VM थे_V_VAUX |_SENT
मेरे_PR_PRP पास_N_NST फिल्म_N_NN को_PSP निर्देशित_N_NN करने_V_VM के_PSP लिए_PSP पैसे_N_NN नहीं_RP_NEG हैं_V_VAUX |_SENT
नदी_N_NN के_PSP पास_N_NST एक_QT_QTC झोपड़ी_N_NN है_V_VB इसलिए_CC हम_PRP वहां_N_NN डेरा_N_NN डाल_V_VM सकते_V_VAUX हैं_V_VAUX !_SENT