Natural Language Processing – Assignment #2

Sequence Tagging and Text Classification

Task #1

1. Yes, the Viterbi algorithm finds the most likely sequence of hidden states that could have generated a sequence of observations. It uses dynamic programming to search through all possible combinations of hidden states and select the sequence with the highest probability of generating the observed sequence. In each iteration, the algorithm calculates the next most likely outcome depending on the outcomes of the previous iteration, thus resulting the most likely sequence of hidden states.
2. Let's consider a scenario of 2 words and 2 tags. We will perform the calculation according to the following matrices where P is the Transition Matrix for the initial state, A is the Transition Matrix and B is the Emission Matrix:

P	Tag1	Tag2
Initial state	0.25	0.75

A	Tag1	Tag2
Tag1	0.65	0.35
Tag2	0.15	0.85

B	Word1	Word2
Tag1	0.05	0.95
Tag2	0.45	0.55

First, we will calculate the values of the C and D matrices according to the correct form of the Viterbi algorithm, given the sequence of [word1, word2]:

C(1,1) = P(1,1) * B(1,1) = 0.25 * 0.05 = 0.0125

C(2,1) = P(1,2) * B(2,1) = 0.75 * 0.45 = 0.3375

C(1,2) = max k of:

[k=1]   C(1, 1) * A(1, 1) * B(1, 2) = 0.0125 * 0.65 * 0.95 = 0.0077

[k=2]   C(2, 1) * A(2, 1) * B(1, 2) = 0.3375 * 0.15 * 0.95 = 0.0481

D(1,2) = argmax k C(1,2) = 2

C(1,2) = max k of:

[k=1]   C(1, 1) * A(1, 2) * B(2, 2) = 0.0125 * 0.35 * 0.55 = 0.0024

[k=2]   C(2, 1) * A(2, 2) * B(2, 2) = 0.3375 * 0.85 * 0.55 = 0.1577

D(2,2) = argmax k C(1,2) = 2

Therefore, the resulting matrices are:

C	Word1	Word2
Tag1	0.0125	0.0481
Tag2	0.3375	0.1577

D	Word1	Word2
Tag1	0	2
Tag2	0	2

And the correct tagging is [tag2, tag2].

Now, we will calculate the values of the C and D matrices according to the modified form of the Viterbi algorithm, given the sequence of [word1, word2]:

C(1,1) = P(1,1) * B(1,1) = 0.25 * 0.05 = 0.0125

C(2,1) = P(1,2) * B(2,1) = 0.75 * 0.45 = 0.3375

C(1,2) = max k of:

[k=1] A(1, 1) * B(1, 2) = 0.65 * 0.95 = 0.6175

[k=2] A(2, 1) * B(1, 2) = 0.15 * 0.95 = 0.1425

D(1,2) = argmax k C(1,2) = 1

C(1,2) = max k of:

[k=1] A(1, 2) * B(2, 2) = 0.35 * 0.55 = 0.1925

[k=2] A(2, 2) * B(2, 2) = 0.85 * 0.55 = 0.4675

D(2,2) = argmax k C(1,2) = 2

Therefore, the resulting matrices are:

C	Word1	Word2
Tag1	0.0125	0.6175
Tag2	0.3375	0.4675

D	Word1	Word2
Tag1	0	1
Tag2	0	2

And the resulting tagging is [tag2, tag1] which is different from the correct tagging.

Task #2

- After observing our confusion matrix from all the datasets, we can see some interesting behaviors.
  First, the biggest confusion was between the classes 1 and 2 which could be caused by the similarity of their respective reviews. The algorithm simply categorizes the reviews based on the scaled frequencies of the words they contain, so reviewers that used rating of 1 or 2 might have used very similar words. Of course, this confusion in classification could easily be made by humans too due to the same reason.
  In addition, we discovered that the model is confused less between classes with larger difference in ratings. For example, our model rarely confuses 1 and 5 but more often confuses between classes 1 and 2 by a big margin.

- After extracting the 15 most valuable words in each dataset, the results are as follows:
  The 15 most valuable words in the "Sports_and_Outdoors" dataset are:
  but, five, four, good, great, love, not, ok, one, perfect, star, stars, three, two, waste
  The 15 most valuable words in the "Pet_Supplies" dataset are:
  but, five, four, great, love, loves, money, not, one, perfect, star, stars, three, two, waste
  The 15 most valuable words in the "Automotive" dataset are:
  but, five, four, good, great, junk, not, one, perfect, star, stars, three, two, waste, works

  Surprisingly, we can see that most of the words are shared between all of the datasets. The main reason behind the phenomenon is that the words simply express the level of satisfaction of a customer from a certain product. For words like "great", "prefect", "waste" the reason is they express the quality of the experience of a product. But words like "five", "four", and the rest of the numbers strongly implies the overall rating of the review.

- After performing cross-domain tests, we conclude that the accuracies may be worse by a few percent, and even be slightly higher in some cases, although not significantly.
  As we saw in the previous bullet, the majority of the most valuable words used by our model are mutual across all datasets. This observation helps us explain why the results of the cross-domain tests were close the results of the traditional tests.
  That is because most of the reviews, regardless of the domain, contain words that are commonly used for describing a general opinion about a certain product without referring to any specific domain property.