1 Viterbi Methodology (Viterbi.ipynb)

The model that I have built here for each step considers 4 cases to calculate the scores: At each point, we are considering the current character and the next character/next possible character and taking their respective probabilities from the bi-gram model.

Case 1: If both the current character let's say 'curr' and the next characters let's call it 'next' are known in the sequence. The score of the sequence of characters at that point would be the log probability of 'next' given 'curr' which we take from the bi-gram distribution. Here we do not need to maximize the score since there is just one score which is the log of the bi-gram probability of 'next' given 'curr' which we add to the score of the model that we had until that point.

Case 2: If the current character 'curr' is known and the 'next' character is masked. In this case, there are multiple cases to take the b-gram probabilities of 'masked'(which is basically 'next') given 'curr' since the character is not known. So, we take all the possible characters that can be replaced in 'masked' and take the log of their bi-gram probabilities. So, it is basically the log of the 'bi-gram' probabilities of 'masked' given curr where 'masked' is all the possible characters. Here we can't take the max score since we do not know what comes next after we have calculated the max-possible scores until this point which here would be for all the possible 'masked' characters given curr which we store and keep for the next step.

Case 3: If the current character is 'masked' and the 'next' character is known. In this case, we know that "masked" has various possible characters which will have a max score for each of the possibilities in our model until that point. So, now that we know 'next' we first find the log probabilities of 'next' given each of the possible "masked" characters and add it to the respective max scores of the characters that we had stored before (i.e if we have to find the log probability of let's say character 'p' given 'a' where 'p' is the known 'next' and 'a' is one of the possible characters of 'curr'/ "masked" we will add the log of that bi-gram probability to the max score of 'a' (one of the possibilities of "masked" given previous which we had stored before in our model where we weren't able to take the max path yet since we don't know until we find a character that we know. So, here we can find the max since all of the previous scores are being added to a known character, so we take the max score in this case until the known character (i.e going on my previous example if the log of the bi-gram probability of 'p' given 'a' added to the previous max score until 'a' which would be "masked" has the max value out of all possible other characters that can go in the "masked" we are going to consider 'a' to be the "masked" character which would give us the maximum probability of the sequence until that point (which would be until 'p' in our example)).

Case 4: If the current character and the 'next' characters both are "masked". So, we have 'masked1', and 'masked2' which I am using for easier interpretation. Here we will be going off of case 2 and case 3. So, as we know "masked1" can be any character from the possible set of values, and "masked2" can be any character from the same set of values. Let's take a few characters as examples for this case: let's say for convenience "masked1" can be the characters from the set a,b,c and "masked2" can be the characters from the same set, so to find the best path until "masked2" we will have to calculate the log of the bi-gram probabilities of 'a' given 'a' added to the max score until character 'a' which would be for "masked1" and 'b' given 'a' added to the max score until character 'b' which would be for "masked1" and so on. The same would be for the log of the bi-gram probability of 'a' given 'b' added to the max score until character 'a' which would be for "masked1" and so on. Here you can notice that we can take the max of the scores until each of the characters in "masked2" and store them (i.e we will be storing the max of the score until 'a', 'b', 'c' which are of "masked2" individually which we can calculate the max of each set of paths going to 'a', 'b' and 'c' respectively being "masked2") and then continue the process further based on what character comes after "masked2". Here as I previously mentioned I have just taken the characters 'a', 'b' and 'c' in my set as an example but in my model I have considered all the possible characters from the bi-gram model.

At each step, the previous total probability until that character is added to the current log probability of the chosen character, and the way I generate scores is by taking the sums of the log probabilities and adding it to the previous path score until that point. When we do this for all the characters in each sentence we get the maximum probability of the sentence under the bigram model's vocabulary using the Viterbi algorithm in polynomial time and space. It would be a polynomial time in my case since I eliminate the choices that would not add to being the maximum if possible at each step rather than checking all possibilities to all possibilities at each step.