IN2110 Oblig 1b:

By Eirik Storrud Røsvik

I had problems getting jupyter notebook to work properly on my home workstation, it would regularly freeze for several minutes then work for a few seconds before freezing again. I therefore finished my code using visual studio code.

Task 1:

a/b):

First the datasets were downloaded (to faster be able to access them) and a pandas dataframe was created to hold this data

The best solution i fount to the first problem, regarding extracting unique symbols, I first created a new dataframe consisting of the IPA column from the initial dataframe. This dataframe were then concatenated to one large string, and then the collections.Counter method creates a list of all symbols and how many times they occur. Last in this problem, a new list is created and all symbols with more than 9 occurrences were added. This resulted in a list of 157 symbols stored as a class object.

For the second problem, extracting the features, a matrix on the form (number if IPA, number of symbols) were created. A loop goes through all the IPA transcriptions, one transcription has one rox in the matrix. Each column represents a symbol, if the symbol is in the transcription, the value 1 is added in the cell corresponding to the row of transcription, column of symbol. The value 0 everywhere else, where the symbols are not in transcription. This matrix is returned from the method

Lastly, the train method was implemented, this method calls the extract feats in order to get the matrix, then fits the matrix and a list of corresponding languages (one per row in the matrix) to the sklearn logistic regression model.

c)

Several evaluating methods were implemented, with the accuracy being 92,7%. The only possible error I could find is that precision seams to always be 1.0.

d)

By saving the model coefficients to a csv file, then exporting them to excel, I was able to produce a heat map to easily be able to see where there are strong and weak weights. By doing this I found that the sound most associated with Norwegian were the sound “ʊ” at 5.97. This sound is used more in Norwegian than any other language, so with more use of this sound it is more likely to be Norwegian.

When it comes to the sounds least likely to result in the model giving Norwegian the sound “a” had the lowest coefficient at -6.303.

Furthermore it appears Norwegian does not have a large usage of any sound, nor are any not used at all. This I presume because for other languages sounds have both much higher and much lower weightings.

Task 2)

e/f)

the get_BIO_sequence method was implemented using a loop to go through the span consisting of elements on the form (x, y, type), every word that begins with “B-“ or “I-“ for the numbers in between the spans, “O” were added, the same was done for the numbers after the last span and the end of the sentence.

g)

Counting the dataset was done as shown in gruppetime 8, the counts were stored as objects in the class. The tokens and labels, where only one of each is stored was stored in lists, the other counts (emission, transition, labels) were stored as dictionaries, with a key and the corresponding occurrences.

h)

For the probabilities, I tried several approaches, mostly trying different ways to structure the dictionaries. The previous attempts have been commented out. The order of iterating through labels and tokens was also experimented with. The current version will not cause errors when used in the Viterbi obtained from Wikipedia/Stack-overflow, but will get a key_error in the Viterbi method supplied with the task. The reason for this is that the back pointer dictionary will map every label to 0, then in the next loop try to use 0 as the key. Due to not having a great understanding of the Viterbi algorithm, I was not successful in restructuring the data to a workable state.

i)

Implementing the Viterbi matrix was not a successful undertaking; I was able to get the code supplied not to crash before the last loop, when the back pointers were used to guess the sentence structure. Using an approach found on Wikipedia, I was able to get some results produced, but they were still not much better than if I randomly guessed. Due to this, I suspect that the error might stem from the probability dictionaries, but all other attempts only resulted in errors. Furthermore, the dictionary structure used now is identical to the one shown in Wikipedia’s example (https://en.wikipedia.org/wiki/Viterbi_algorithm)