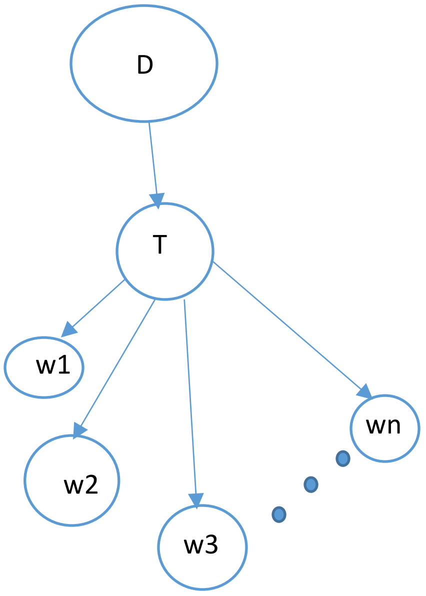
Assignment 4 Part 2 – Report

1) Problem Formulation –

The solution to the problem was formulated using a Naïve Bayes Classifier which allocates topics to documents based on a distribution over calculated posteriors.

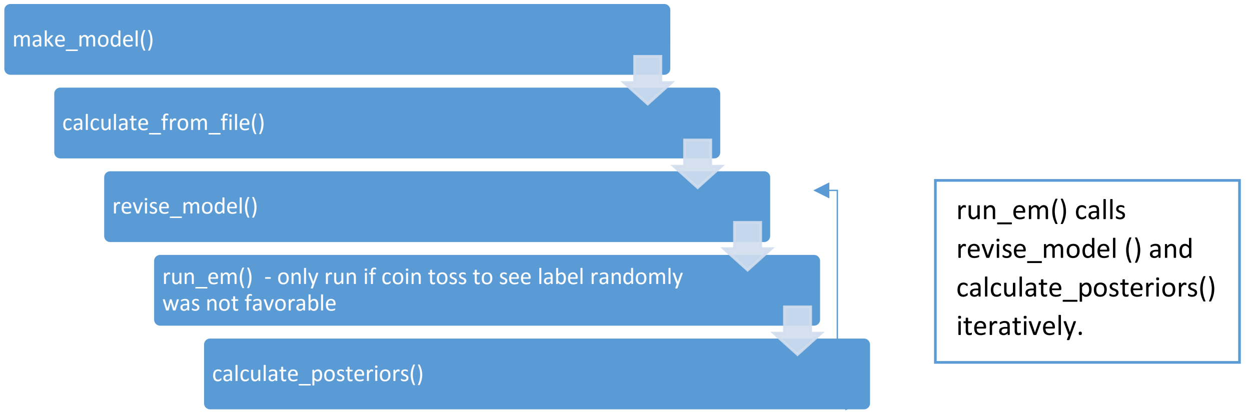
Following is the **Bayes Net** –



Hence the posterior for every document was calculated as - P(T|W1,W2,..Wn) = (P(W1,W2,…Wn|T) X P(T))/ P(W1,W2,W3..Wn) where W1,W2,W3..Wn are words belonging to a particular document under a given topic T.

Therefore, with the Naïve Bayes Assumption – P (W1, W2, …Wn|T) = P (W1|T) X P(W2|T) …. P (Wn|T) and so the calculation with this independence assumption becomes simplified.

2) Program Description - Here's a call flow of the functions in the program for training the model



make_model()

calculate_from_file()

revise_model()

run_em() - only run if coin toss to see label randomly was not favorable

calculate_posteriors()

run_em() calls revise_model () and calculate_posteriors() iteratively.

Fig. - Model training functions - a call flow

1) **make_model ()** – Traverses through the directory structure, opening each file in the tree and sends the filepath to calculate_from_file()

2) **calculate_from_file()** – Parses each file sent by the make_model() function and calculates associated word frequency and records it in the frequency dictionary.

3) **revise_model()** – Revises the frequency dictionary to include corresponding word likelihoods given the topic.

4) **run_em() –** Runs the EM Algorithm on the frequency dictionary in the case where there are randomly assigned labels due to the coin toss comparison with the fraction, and which hence means the topics are missing. It calls revise_model() and calculate_posteriors() iteratively to help classify the data with the missing topics.

5) **calculate_posteriors() -** Calculate the posteriors of the topics and updates the dictionary to reflect the same with the corresponding data from documents – classifying them as it goes.

6) **predict_topic() –** This is the function involved in calculating posteriors and estimating topics for **test** data.

Assumptions/Design Decisions /Problems –

1) The user will input values in correct datatype. Checks can be placed to make sure to alert user to enter arguments in the correct format and order.

2) The posterior of a topic given a word which is absent in the model is assumed to be a very small probability value of 0.0000001.

3) The assumption to the total number of topics in the test directory, which are going to be 20. (Given in question)

4) The EM algorithm was run only for 2 iterations as with more iterations the program execution time increased.

5) The assumption that model file does not need to be human readable. The model file was hence written using pickle.

6) Assumption that the train and test directories the user inputs contain immediate sub-directories which are topic names and which further contain files. The directory hierarchy tree was assumed to be of a single level and the sub-directory names were assumed to be topics/labels for the data.

Outputs on test data with different values of the fraction

1) fraction = 0, Accuracy = 5%

2) fraction = 0.2, Accuracy = 14%

3) fraction = 0.3, Accuracy = 27%

3) fraction = 0.5, Accuracy = 52%

4) fraction = 1, Accuracy = 70%