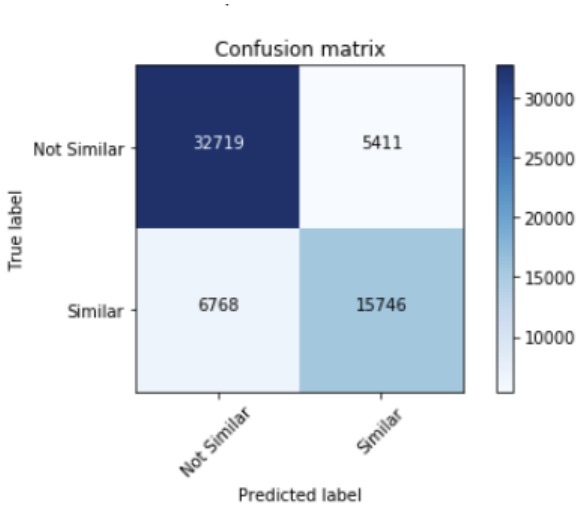
1) I had used the 15% split of the training data for testing and got an accuracy of about 79% on it.

   Following is the confusion matrix:



   About 30% of similar sentences are classified as not-similar.

2) I implemented the following things for pre-processing the text:
   i) Replacing "'s" with " is"
      This was so that words like "what's" are converted to 'what is'

   ii) Removing stop words
      This was to remove words like 'the' 'have' which occur too often in sentences and might trick the network into thinking that two sentences are same, even if they are different.

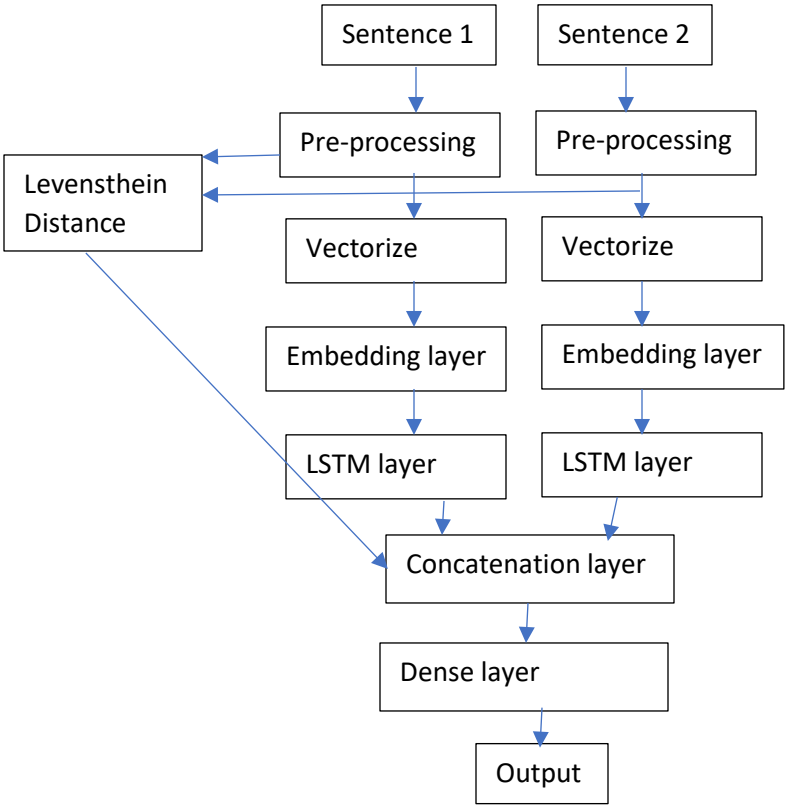
   iii) Removing punctuations
      This was because two similar sentences written in different ways might have different punctuations.

   iv) Converting to lower case
      This was because changing the words from upper to lower case does not change their meaning.

   v) Lemmatizing the words
      This was so to bring all the words to their fundamental form.

3)
   a) Initialize all the elements of the embedding matrix to zero
   b) Run a loop for the indices and words in the tokenizer
   c) Convert the current word to its vector form using the word2vec model, if the word exists in the model's vocabulary

4) The model architecture is as follows:



5) The training was painfully slow.