CS7.401: Introduction to NLP | Assignment 3 report

Aaditya sharma,2019113009

Q1)

1.1)

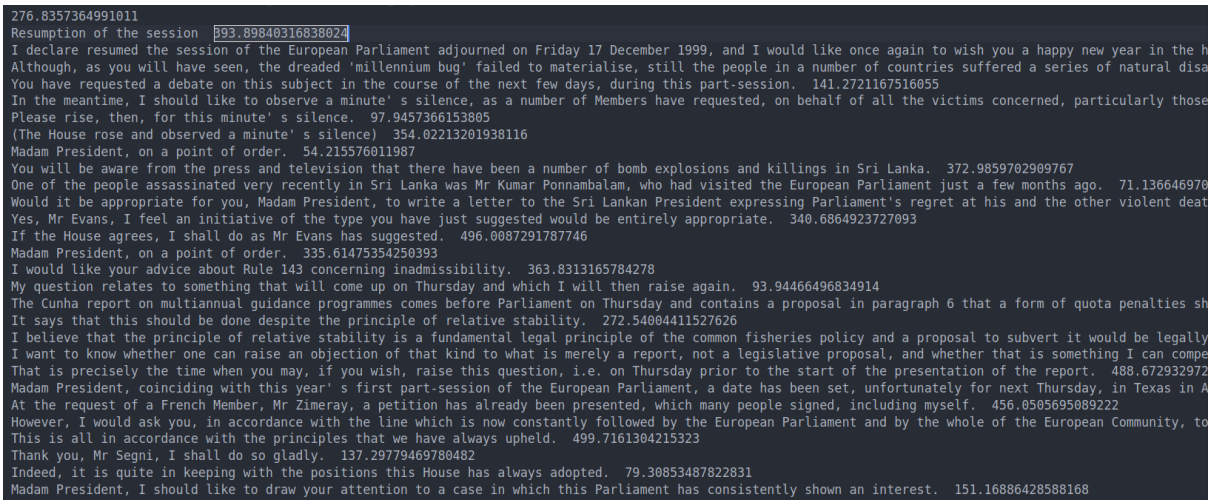
Created a neural network model using LSTM.
Steps ->

- I read the data and tokenized it.
- Appended “<S>” at the start of every sentence and “<E>” at the end of every sentence.
- Created a list of unique tokens.
- Then I created a word to index and index to word dictionary for the given dataset.
- Then I initialized my model using ->
  model = Sequential()
  Took input_dim of Embedding layer = len(unique_tokens)
  Output_dim = 25
  Input_length = 4
  Compiled using adam compiler
- Then I trained my model using *model.fit()* and sent data in batches of 5000.
- In *model.fit()* parameters are x,y,ephochs , where x is context(n-1 gram) , y is obtained by doing the *one-hot encoding* of the target(last term of n gram) and epochs = 10
- I saved my model in a directory called modelno1 and I load the model every time to give the probability of any input sentence using the file *language_model.py*

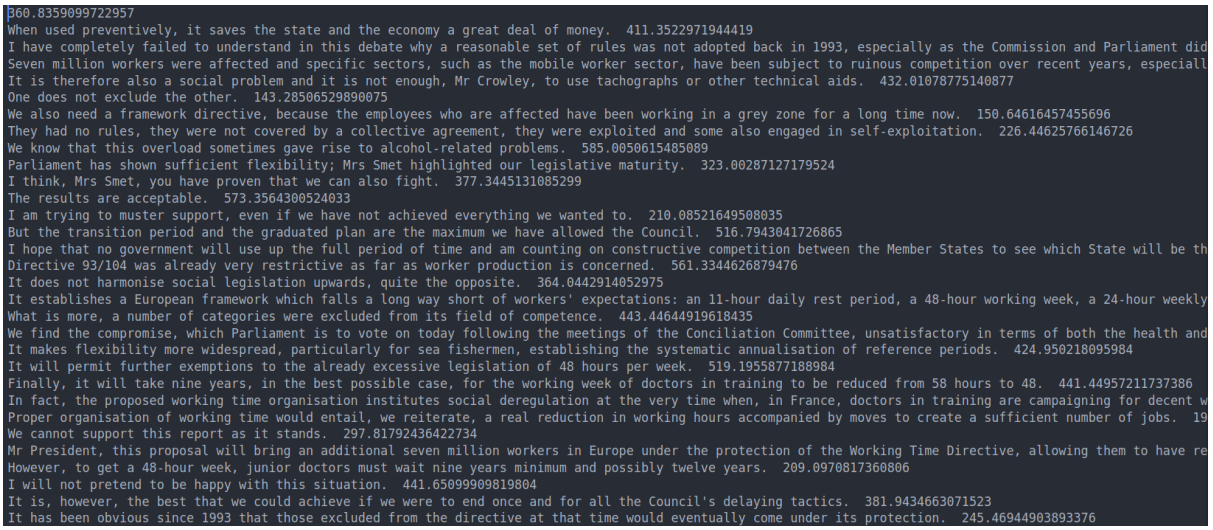
1.2)

To get the perplexity score I used my model (*model.predict(x)*) which took each sentence as a parameter and gave the probability for each sentence.
Then from the probabilities get the perplexity score (1/p^1/n)

For train data avg perplexity = 276.8357364991011 ->



For test data avg perplexity = 360.8359099722957->



Q2)

2.1)

Created a sequence to sequence model.

Steps ->

- I read the data (both the french and the english dataset) and tokenized it.

- Created a list of unique tokens for both datasets.
- I calculated the max length of the English dataset and then made each sentence of the dataset equal to that length and did a similar thing for the french dataset.
- Then I created a word to index and index to word dictionary for both the datasets.
- Then I initialized my model using ->
  model = Sequential()
  Took input_dim of Embedding layer = len(unique_tokens of english dataset)
  Output_dim = 15
  Input_length = max_len_of_english
  Mask_zero = True
  RepeatVector parameter = max_len_of_french
- Compiled using adam compiler.
- Then trained my model with model.fit(x,y,ephochs) with batch size = 100.
- Parameter x is the english sequence , y is obtained by doing one-hot encoding and epochs = 10.
- I saved my model in a directory called model2 and I load the model every time (with word to index and index to word dict) to get the translation of any input sentence using the file machine_translation*.py*

2.2)

Then load the model and send the data line by line and get the translation of each line . After getting the translation of each line use - *nltk.translate.bleu_score.sentence_bleu* to get the bleu score of each sentence and *nltk.translate.bleu_score.corpus_bleu* to get the corpus bleu score of all the sentences.

Do it for both M1 model and M2 model.
Got M2 by using weights from the model trained on the English for the Encoder and French task for the Decoder and using this fine-tuned the Encoder and Decoder on the parallel corpus for translation.

For M1 model ->

For train dataset corpus-level BLEU score =
5.312458432322297e-168->

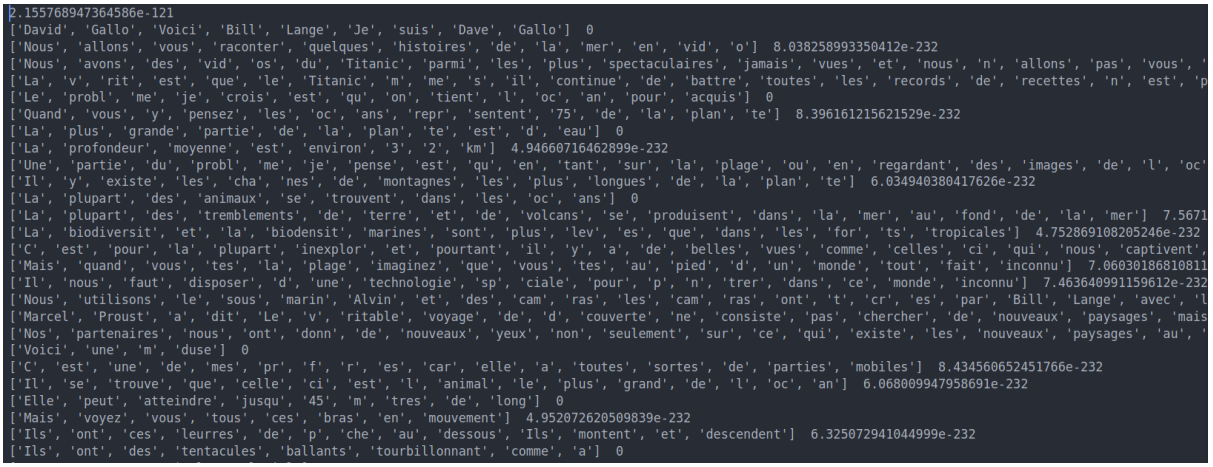
```
5.312458432322297e-168
['BF:', ':', ':', 'que', 'de', 'de']  0
['Et', 'nous', 'que', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', '?', '?']  9.418382295637229e-232
['Nous', 'nous', 'nous', "n'a-t-il", 'de', 'de', 'de', 'et', 'et', 'que', 'que', 'que', 'que', 'que', 'que', 'de', 'de', 'de', 'Goûtez', 'sang', '!']  1.0182922
['Mais', 'y', 'a', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'hanche.', '<title>Maira', 'Dunham-Jones', ':', 'Ré
['Et', 'ne', 'que', 'pas', 'que', 'que', 'pas', 'pas', 'de', 'de', 'de']  0
['<title>Andreas', 'Raptopoulos:', ':', ':', ':', 'de', 'de', 'de', 'ouvrières']  6.034940380417626e-232
['<title>Mark', 'Shaw', ':', 'Rénover', 'nait.', 'imperméable</title>']  0
['<title>Maira', 'Kalman,', '"Kiteflyer\'s', '"Kiteflyer\'s']  0
['Et', 'il', 'a', 'de', 'de', 'de', 'que', 'que', 'que', 'que', 'article', 'article', 'Journal', 'Journal', 'Journal', 'Médecine', 'Médecine', 'Et', 'une', 'la'
['<title>Jennifer', 'Killingsworth:', 'Lee', "l'ingéniosité", 'de', 'de', 'de', 'de', 'de', 'la']  3.865846993487796e-155
['Les', 'Sissay', 'des', 'le', 'de', 'de']  5.9764165035645784e-232
['<title>Nate', 'Iwasa:', ':', 'JR', 'vœu', 'vœu', 'Prix', 'Prix', 'Prix', 'de', 'et', 'et', 'fois...', 'fois...', 'hypothèse</title>']  8.432010639666965e-232
['Les', 'avons', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de']  0
['Et', 'nous', 'a', 'de', 'de', 'de', 'de', 'de', 'et', 'et', 'et', 'et', 'et', 'et', 'et', 'et', 'et', 'et', 'de', 'de']  3.3864995705940004e-155
['Mais', 'Hypponen:', ':', 'vous', 'vous', ':', ':', ':', ':', ':', ':', 'que', 'que', 'que', 'de', 'de', '<speaker>Nic', 'Marks</speaker>']  1.2508498911928379
['Nous', 'avons', 'les', 'les', 'annuel', 'de', 'de', 'de', 'de', 'de', 'CO2?']  0
['Nous', 'Besser', 'les', 'les', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'et', 'et', 'et', 'et', 'et', 'de', 'analysés', 'analysés']  9.488382896503462e
['Et', ':', 'a', 'a', 'de', 'de', 'de', 'de', 'et', 'et', 'et', 'et', 'et', 'et', 'a', 'et', 'et', 'et', 'la']  9.997801362989555e-232
['Mais', 'y', 'a', 'soucie', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'Tosca', 'Tosca', 'de', 'Nous', 'avons',
['Beijing']  0
['Alors', 'la', 'souhaitons', 'de', 'de', 'de', 'de', 'primitifs,', 'de', 'de', 'de']  6.8949282624564e-232
['Et', 'la', 'de', 'de', 'de', 'de', 'de', 'de', 'de', "l'humanité</title>"]  5.622528097505664e-232
['Comment', 'la', 'les', 'de', 'de', 'de', 'société?']  8.416851712392762e-232
['<title>David', 'Dunham-Jones', ':', 'Rénover', 'le', 'banlieues</title>']  0
['Et', 'la', 'de', 'de', 'de', 'de', 'de', 'de', 'de']  6.744160953836975e-232
['Comment', 'étape:', 'des', 'le', 'nait.', 'sacrifier?']  8.34076112986429e-232
['Comment', 'que']  0
```

For test dataset corpus-level BLEU score =
3.283495847334756e-220->

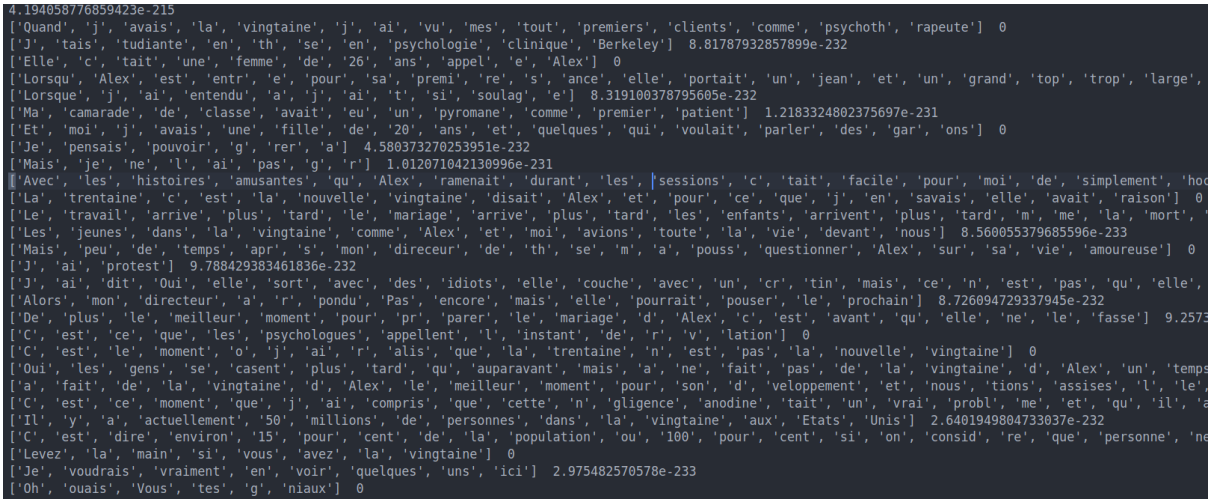
```
3.283495847334756e-220
['Et', 'je', 'que', ':', ':', 'que', 'que', 'que', 'pas', 'pas', 'de']  0
['Comment', 'Lomborg', 'que', 'de', 'de', 'de', 'de']  5.483065946899793e-232
['<title>Alex', 'Fitzgerald:', 'Miracle', 'jonglent', 'nait.', 'l’univers.</title>']  0
['<title>Mikko', 'Hypponen:', 'nous', 'la', 'de', 'de', 'de', 'et', 'et', 'et', 'de', 'de', 'de', 'et', 'et', 'et', 'de', 'de', 'de', '<keywords>talks,', 'Strin
['Comment', 'que', ':', ':', ':', 'de', 'de']  0
['<title>Gary', 'Pollan', ':', 'les', 'de', 'Stephen', 'de', 'plante</title>']  6.085166479973199e-232
['Comment', ':', 'Lee', ':', 'de', 'de', 'de', 'de']  7.445183326929416e-232
['Je', ':', 'que', 'que']  0
['Mais', 'ne', 'pas', 'pas', 'pas', 'nuit-là']  0
['Il', 'que', 'a', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'Quid', 'invente']  1.1200407237786664e-231
['Alors', 'il', 'chauffeur', ':', 'taxi', 'demandait,', 'de', 'de', 'demandaient,', 'demandaient,', 'de', 'de']  0
['<title>Mikko', 'Hypponen:', ':', 'de', 'de', 'de', 'trahi', 'trahi', 'Prix', ':', 'de', 'de', 'de', 'de', 'réagir.</title>']  9.33155372164043e-232
['Je', 'que', 'que', 'que', 'de', 'de', 'de']  3.0963867043779474e-232
['<title>Janet', 'Jorgensen', ':', ':', 'Biohacking', 'Prix', 'Prix', 'Prix', 'affecte-t-elle', 'affecte-t-elle', 'de', 'de']  2.5802019721284275e-232
['Je', 'est']  0
['Je', 'je', 'vous', 'que', 'que', 'que', 'Loony', 'Tunes', 'Tunes', 'de', 'de', 'je', 'je', 'je', 'je', 'poursuit', 'que', 'de', 'Bip', 'Bip', 'Bip', '<talkid>
['Et', 'je', 'que', 'pas', 'que', 'que', 'que', 'que', 'que', 'que', 'que', 'pas', 'de']  3.811845339571429e-232
['Et', 'la', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de']  6.517294086106268e-232
['<title>Maira', 'Kalman,', 'Miracle', 'jonglent', 'nait.', 'l’univers.</title>']  0
['Et', 'pourquoi', 'que', 'que', 'que', 'que', 'pas', 'pas', 'pas', 'futés', '?']  0
['Comment', 'une', 'une', 'une', 'à', 'de', 'de', 'Tunes', 'Tunes', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de']  1.1640469867513693e-231
['Et', 'nous', 'a', 'de', 'de', 'à', 'de', 'de', 'la', 'de', 'de', 'de', 'la', 'la', 'de', 'de', 'de']  8.972141065609098e-232
['Et', 'je', 'je', 'vous', 'vous', 'Loony', 'Loony', 'Tunes', 'Tunes', 'Tunes', 'vous', 'pathétique', 'pathétique', 'que', 'que', 'que', 'de', 'de', 'Bip', 'Bip
['Les', 'Tulley', ':', 'les', 'de', 'de', 'de', 'de', 'plante</title>']  0
['Nous', 'avons', 'a', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', 'de', "Qu'en", 'invente
['Comment', 'Seung:', 'diffuser', ':', "n'utilises"]  0
['Je', 'pourquoi', 'que', 'que']  0
```

For M2 model ->

For train dataset corpus-level BLEU score =
2.155768947364586e-121->



For test dataset corpus-level BLEU score = 4.194058776859423e-215->



We got a very small bleu score for most of the sentences and the reason behind this is that due to computational barriers we can not train our model on a lot of epochs and to get considerable it will take a **LOT** of time. So the model which I am submitting is not very good but if trained for a long time will do its job nicely.