NLP REPORT
ASSIGNMENT 2

NAVIDHA JAIN:
B-TECH 2020223
MANAV SAINI:
B-TECH 2020518

Outcomes:
Part A:

a) Please refer to convert.text in the zip file.
b) Top 4 bigrams:

	PREV	WORD	COUNT
5177	gon	na	65.5
1982	good	morning	39.5
5176	wan	na	38.5
9795	cant	wait	34.5

c) Accuracy of Dataset A is `0.9254658385093167`

Part B:

a)

- Pre-processing the text of A1 dataset was for the purpose of the following using the python library ‘nltk’ and regex:
  - Lowercasing of each word
  - Removal of extra spaces
  - Removal of websites and HTML tags
  - Removal of mentions starting with “@”
  - Removal of hashtags
  - Removal of punctuations
  - Removal of stopwords
  - Spelling correction
- Formation of unigram dictionary
  - Key: Word; Value: Count
- Formation of nested bigram dictionary

- (Let's say our bigram is A B)
- Key: Word(B); Value: Dictionary
- Sub dictionary
  - Key: Word before(A); Value: Count
- Add-k (Laplace) smoothing
  - We have taken k to be 0.5
  - Assigned the value of 0.5 to the unseen words
  - Increased the count of seen words by 0.5

- Calculation of Probability

```
string=word+' '+prev
sent=sentiment_scores(string)
bigram_prob[word][prev]=((bigram_dict[word][prev])/
(unigram_dict[prev] + 0.5*v))*sent[0] + sent[1]
```

  - Beta factor
    - We have used Vader Sentiment library to calculate out beta factor
    - The beta factor used in our equation is divided into the two following parts:
      - sent[0]
        - -1 for negative and +1 for positive sentiment oriented sentences
        - This ensures that in sentence formation, done for the 500 generated samples, if the first randomly chosen word is positive then preference will be given to words with positive sentiment value generating an overall positive sentiment oriented sentence and similarly, if the first randomly chosen word is negative then preference will be given to words with negative sentiment value generating an overall negative sentiment oriented sentence.
        - This part gets multiplied with the actual probability and is used to distinguish between positive words and negative words.
      - sent[1]

        - This part ensures that words that have a vader score of more than 0.05 or less than 0.05 get preference in sentence formation for positive sentiment oriented sentence and negative sentiment oriented sentence respectively.
        - This part gets added in the equation to deal with the preference of words according to how positive or negative they are according to their Vader score.
    - Correctness of Beta factor
        - Our beta factor is ensuring bifurcation of positive and negative sentiment oriented bigram words and is giving preferences to those which have a higher vader score thus ensuring overall positive and negative sentiment oriented sentence formation.

b) Please refer to the generated_data csv file in the zip file submitted.

c) Average Perplexity of the 500 generated samples is `262.21022912939577`

Perplexity has been calculated as $P(\text{sentence})^{-1/(\text{length of the sentence})}$

d) The 10 generated samples are:

POSITIVE SENTENCES:

better hope great time law wedding wow thats whats going aunt
feel better hope your right really don't think anyone heal upper
really don't think is gon na go back work listening wishing
get thanks really don't feel good time got new pack gum
is gon na go shopping totally didn't either way don beloved
yeah thanks really dont feel better hope great time study review

NEGATIVE SENTENCES:

think is gon na go back work don't really busy we
would totally didn't know right really don't feel good wondering local
cant stop totally didn't know right really don't feel like summary
right really don't feel like said wow stunning alas nowhere kind
didn't get thanks really dont think is still can't sleep cheesecake
didn't think is gon na go back work don't really volunteer

e) Accuracy of Dataset B is `0.9301242236024845`
(Clearly, Accuracy of Dataset B is greater than Dataset A.)

Source reference: For Model Training we have used the code provided to us in the Assignment PDF.