ASSIGNMENT 2

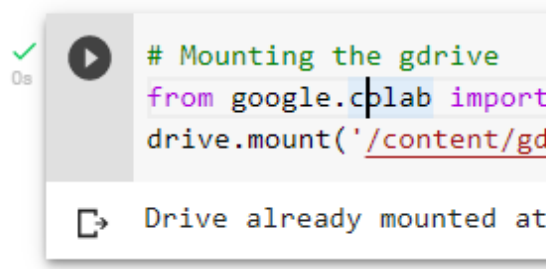
- Manvi Goel(2019472) and Jahnvi Kumari(2019469)

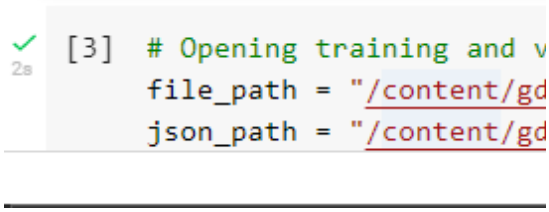
Execution

- Mount the drive.

```
# Mounting the gdrive
from google.colab import drive
drive.mount('/content/gdrive')
```

- Open the training and the validation files.
- Run the cells in the set order, or run all the cells





Bigram Model

Methodology

Without Smoothing

- Calculate the number of times the previous word occurs.
- Calculate the number of times the bigram for all the options occurs.
- Choose the word with the maximum probability.
- If all words have 0 probability, choose the last word
- If two words have the same probability, choose the first out of the two.
- The function is called = probabilitySimple.

- Accuracy (Without Future Context) :

```
The accuracy of the bigram model wihout smoothing is:  47.15
```

- Accuracy (With Future Context)

```
The accuracy of model without smoothing:  53.0
```

Laplace Smoothing

- Calculate the probability by adding one to the count of occurrences for the bigram of option and previous word.
- Add the total number of words in the vocabulary to the count of occurrences of the previous word to normalize.
- If two words have the same probability, choose the first out of the two.
- The function is called = probabilityLaplace.

- Accuracy (Without Future Context)

```
The accuracy of the bigram model with Laplace smoothing is:  47.2
```

- Accuracy (With Future Context)

```
The accuracy of model with Laplace Method:  54.2
```

Add K Smoothing

- Calculate the probability by adding k (less than 1 and greater than 0) to the count of occurrences for the bigram of option and previous word.
- Add the total number of words in the vocabulary to the count of occurrences of the previous word to normalize.

- If two words have the same probability, choose the first out of the two.
- The function is called = probabilityAddK.
- k is chosen as 0.2. The value of k does not make a serious difference in the validation set accuracy due to a large number of words in the vocabulary. The k's tried are 0.1, 0.0001, 0.5 and 0.9. The probability is normalized. The value of k can be changed in the findMaskedWord function.
- Accuracy (Without Future Context)

```
The accuracy of the bigram model with Add k smoothing is:  47.2
```

- Accuracy (With Future Context)

```
The accuracy of model with Add k method:  55.45
```

Preprocessing

- Convert all the words to lower case in the sentence.
- Remove all the punctuations like full stop, comma.
- Regexp tokenizer for words is used.
-

Parameters

- The value of k.
- The method of smoothing. (None, Laplace, Add k)
- The current function uses the validation set to calculate the accuracy. Upload the test set, and update the file path to check the accuracy of the test set.

```
[3]  # Opening training and validation files.
     file_path = "/content/gdrive/My Drive/CBT LM Dataset/train.txt"
     json_path = "/content/gdrive/My Drive/CBT LM Dataset/validation.jsonl"
     file = open(file_path,'r')
     text = file.read()
     file.close()
```

Assumptions

- The test file is in json.
- The test sentences are tokenized.
- The previous words are present in the vocabulary.
- In case of the same probability, the first option will be chosen
- In all options with 0 probability (without smoothing), the last option is selected.
- The co-occurrence matrix can be calculated using the function //. It is not calculated due to the lack of space and the time taken during its computation.
- An alternative to the co-occurrence matrix is the bigram counter in the model, which stores the count of bigram per word in a nested dictionary.
- The sample outputs are given in the output file.
- The complete vocabulary is considered.

Bonus

Future Context
The word next to the masked word is considered for the future context.
The reason for choosing one word for future context is:

- By selecting one word, the bigram functions can be recycled without creating new functions.
- Using a one-word in advance reduces the chance of reaching the end of the sentence.
- Choosing multiple words would increase the computational cost.
- We can see a satisfactory level of increase in the accuracy for the three models.

Methodology

- Calculate the probability of the next word given option using the bigram model for all the options
- Multiply the probability of the next word given the option to the probability of the option given the previous word.
- Choose the word with the highest probability.
- The functions for the three are probabilitySimpleFuture and findMaskedWordFuture.

Thank You