Team members

1. Sai Hruthik Gangapuram
2. Sujith Kumar Gajarla

Task 1: Importing packages

In [1]:
```
import re
import torch
import random
import pandas as pd
import torch.nn as nn
from torch.utils.data import Dataset
from torch.utils.data import DataLoader
from torch.nn.utils.rnn import pad_sequence
from sklearn.model_selection import train_test_split
device = 'cpu'
```

Task 2: Data Loading

In [2]:
```
data_frame = pd.read_excel('dataset.xlsx')
data_frame.head(6)
```

Out[2]:

	English	Hindi
0	Yale offers advanced degrees through its Gradu...	येल अपने ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज ...
1	Browse the organizations below for information...	अध्ययन के कार्यक्रमों, शैक्षणिक आवश्यकताओं और ...
2	Graduate School of Arts & Sciences.	ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज।
3	Yale’s Graduate School of Arts & Sciences offe...	येल के ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज एम...
4	School of Architecture.	स्कूल ऑफ आर्किटेक्चर।
5	The Yale School of Architecture’s mandate is f...	येल स्कूल ऑफ आर्किटेक्चर का जनादेश प्रत्येक छा...

Task 3: Data preprocessing

1. Word to Index
2. Index to word
3. Word counts
4. Normailizing the sentence

In [3]:
```
START_TOKEN = 0
END_TOKEN = 1

class Language:
    def __init__(self, name):
        self.language_name = name
        self.word_to_index = { "START": START_TOKEN, "END": END_TOKEN }
        self.word_to_count = {}
        self.index_to_word = { START_TOKEN: "START", END_TOKEN: "END" }
        self.num_words = 2  # Count START and END tokens

    def add_sentence(self, sentence):
        for word in sentence.split(' '):
            self.add_word(word)

    def add_word(self, word):
        if word not in self.word_to_index:
            self.word_to_index[word] = self.num_words
            self.word_to_count[word] = 1
            self.index_to_word[self.num_words] = word
            self.num_words += 1
        else:
            self.word_to_count[word] += 1
```

In [4]:
```
def normalizeString(sentence):
    sentence = sentence.lower().strip()
    sentence = sentence.replace('\xa0', ' ')
    sentence = re.sub(r"([,.!?])", r" \1", sentence)
    sentence = re.sub(r"[.!?]+", r"", sentence)
    return sentence
data_frame['English'] = data_frame['English'].apply(lambda sentence: normalizeString(sentence))
```

```
data_frame['Hindi'] = data_frame['Hindi'].apply(lambda sentence: normalizeString(sentence))
data_frame.head(5)
```

Out[4]:

	English	Hindi
0	yale offers advanced degrees through its gradu...	येल अपने ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज ...
1	browse the organizations below for information...	अध्ययन के कार्यक्रमों , शैक्षणिक आवश्यकताओं और...
2	graduate school of arts & sciences	ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज।
3	yale’s graduate school of arts & sciences offe...	येल के ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज एम...
4	school of architecture	स्कूल ऑफ आर्किटेक्चर।

In [5]:

```
def read_languages(data_frame):
    pairs = [list(lang_pair) for index, lang_pair in data_frame.iterrows()]
    input_lang = Language('English')
    output_lang = Language('Hindi')
    return input_lang, output_lang, pairs
```

In [6]:

```
def preprocess(data):
    source_lang, target_lang, sentence_pairs = read_languages(data)
    print("Read %s sentence pairs" % len(sentence_pairs))
    print("Counting words...")
    for pair in sentence_pairs:
        source_lang.add_sentence(pair[0])
        target_lang.add_sentence(pair[1])
    print("Counted words:")
    print(source_lang.language_name, source_lang.num_words)
    print(target_lang.language_name, target_lang.num_words)
    return source_lang, target_lang, sentence_pairs

source_lang, target_lang, sentence_pairs = preprocess(data_frame)
print(random.choice(sentence_pairs))
```

```
Read 129 sentence pairs
Counting words...
Counted words:
English 533
Hindi 598
['yale school of medicine graduates go on to become leaders in academic medicine  ', 'येल स्कूल ऑफ मेडिसिन के स्नातक शैक्षणिक चिकित्सा में अग्रणी बन जाते हैं।']
```

In [7]:
```
source_lang.index_to_word[21]# index to word example in input language
```

Out[7]:
```
'organizations'
```

In [8]:
```
target_lang.index_to_word[89]# index to word example in output language
```

Out[8]:
```
'ने'
```

Task 4 : Creating Custom Dataset

In [9]:
```
class CustomDataset(Dataset):

    def __init__(self, df):
        self.df=df

    def __len__(self):
        return len(self.df)

    def indexesFromSentence(self, lang, sentence):
        return [lang.word_to_index[word] for word in sentence.split(' ')]

    def tensorFromSentence(self, lang, sentence):
        indexes = self.indexesFromSentence(lang, sentence)
        indexes.append(END_TOKEN)
        return torch.tensor(indexes, dtype=torch.long, device=device)

    def __getitem__(self ,idx):
        languages = self.df.iloc[idx]
        input_tensor = self.tensorFromSentence(source_lang, languages['English'])
        target_tensor = self.tensorFromSentence(target_lang, languages['Hindi'])
        return input_tensor, target_tensor, languages['English'], languages['Hindi']
```

Task 5: Spliting the dataset into training | testing| validation

In [10]:
```
training_data, testing_data = train_test_split(data_frame, test_size=0.2, random_state=42)

validation_data, testing_data = train_test_split(testing_data, test_size=0.5, random_state=42)
```

In [11]:
```
train_data_set = CustomDataset(training_data)
valid_data_set = CustomDataset(validation_data)
test_data_set = CustomDataset(testing_data)
```

In [12]:
```
print('Size of Training dataset: {}'.format(train_data_set.__len__()))
print('Size of Testing dataset: {}'.format(test_data_set.__len__()))
print('Size of Validation dataset: {}'.format(valid_data_set.__len__()))
```

```
Size of Training dataset: 103
Size of Testing dataset: 13
Size of Validation dataset: 13
```

In [13]:
```
train_data_set[50]# sample
```

Out[13]:
```
(tensor([368,  78, 344, 369, 164, 366,  42, 370, 371,  18,   1]),
 tensor([420, 258, 217,  30, 195, 413,  53, 421, 251, 358, 171,   1]),
 'we have been expanding international collaborations in many areas ',
 'हम कई क्षेत्रों में अंतरराष्ट्रीय सहयोग का विस्तार कर रहे हैं।')
```

Task 6: Loading dataset into Batches

In [14]:
```
def collate_fn(batch):
    batch = sorted(batch, key=lambda x: len(x[0]), reverse=True)
    input_seqs, target_seqs, input_language, out_language = zip(*batch)
    # Pad the input sequences with zeros
    padded_input = pad_sequence(input_seqs, batch_first=True)
    # Pad the target sequences with zeros
    padded_target = pad_sequence(target_seqs, batch_first=True)
    return padded_input, padded_target, input_language, out_language
```

In [15]:
```
train_loader = DataLoader(train_data_set, batch_size=8, shuffle=True, collate_fn=collate_fn)

val_loader = DataLoader(valid_data_set, batch_size=8, shuffle=True, collate_fn=collate_fn)

test_loader = DataLoader(test_data_set, batch_size=8, shuffle=True, collate_fn=collate_fn)
```

In [16]:
```
print('Total number of batches in train data loader: {}'.format(len(train_loader)))
print('Total number of batches in test data loader: {}'.format(len(test_loader)))
print('Total number of batches in validation data loader: {}'.format(len(val_loader)))
```

```
Total number of batches in train data loader: 13
Total number of batches in test data loader: 2
Total number of batches in validation data loader: 2
```

Task 7: Displaying 1st sample in each batch

Train data loader

In [17]:
```
for batch_index, packed in enumerate(train_loader):
    input_tensors, output_tensors, input_language, out_language = packed
    print("\033[1mTraining Batch number-----> {}\033[0m".format(batch_index+1))
    # print the first input and output tensors along with their respective languages
    print("Input Language:", input_language[0])
    print("Input Tensor Shape:", input_tensors[0].shape)
    print("Input Tensor:", input_tensors[0])

    print("Output Language:", out_language[0])
    print("Output Tensor Shape:", output_tensors[0].shape)
    print("Output Tensor:", output_tensors[0])
    print('-----------------------------------------------------------------------------------------------------------')
    print("\n")
```

```
Training Batch number-----> 1
Input Language: the yale school of art has a long and distinguished history of training artists of the highest caliber
Input Tensor Shape: torch.Size([20])
Input Tensor: tensor([20,  2,  9, 10, 58, 59, 37, 60, 14, 61, 62, 10, 63, 64, 10, 20, 65, 66,
        18,  1])
Output Language: येल स्कूल ऑफ आर्ट में उच्चतम क्षमता के प्रशिक्षण कलाकारों का एक लंबा और विशिष्ट इतिहास है।
Output Tensor Shape: torch.Size([23])
Output Tensor: tensor([ 2,  5,  6, 67, 30, 68, 69, 14, 70, 71, 53, 57, 72, 10, 73, 74, 21,  1,
         0,  0,  0,  0,  0])
-------------------------------------------------------------------------------------------------------------


Training Batch number-----> 2
Input Language: yale is known for its residential college system , which provides students with a supportive community and numerou
s opportunities for social and intellectual engagement
Input Tensor Shape: torch.Size([26])
Input Tensor: tensor([  2,  48, 460,  23,   7, 506, 277, 507,  28, 254, 508, 142, 211,  37,
        509, 171,  14, 230, 137,  23, 178,  14, 510, 377,  18,   1])
Output Language: येल अपनी आवासीय कॉलेज प्रणाली के लिए जाना जाता है , जो छात्रों को एक सहायक समुदाय और सामाजिक और बौद्धिक जुड़ाव के कई अवसर प्र
दान करता है।
Output Tensor Shape: torch.Size([29])
Output Tensor: tensor([  2, 324, 572, 316, 573,  14,  32, 521, 245, 106,  24, 102, 261,  37,
         57, 574, 308,  10, 114,  10, 575, 427,  14, 258, 163,  19,  20,  21,
          1])
--------------------------------------------------------------------------------------------


Training Batch number-----> 3
Input Language: yale’s international research , teaching , and learning activities are undertaken in a wide variety of centers and
programs across all academic fields
Input Tensor Shape: torch.Size([25])
Input Tensor: tensor([ 33, 164,  32,  28, 186,  28,  14, 187, 188, 144, 189,  42,  37, 190,
        191,  10, 192,  14,  26, 193, 194,  29, 195,  18,   1])
Output Language: येल के अंतरराष्ट्रीय अनुसंधान , शिक्षण और सीखने की गतिविधियां सभी शैक्षणिक क्षेत्रों में विभिन्न प्रकार के केंद्रों और कार्यक्रमों में की जाती हैं।
Output Tensor Shape: torch.Size([25])
Output Tensor: tensor([  2,  14, 195,  28,  24, 213,  10, 214, 129, 215, 216,  25, 217,  30,
        218, 219,  14, 220,  10,  23,  30, 129, 221, 171,   1])
--------------------------------------------------------------------------------------------


Training Batch number-----> 4
Input Language: the university has a rich athletic tradition , with 35 varsity sports teams and numerous club and intramural sport
s
Input Tensor Shape: torch.Size([21])
Input Tensor: tensor([ 20, 207,  59,  37, 511, 512, 513,  28, 211, 514, 515, 516, 517,  14,
        230, 396,  14, 518, 516,  18,   1])
Output Language: विश्वविद्यालय की एक समृद्ध एथलेटिक परंपरा है , जिसमें 35 विश्वविद्यालय खेल टीमें और कई क्लब और इंट्राम्यूरल खेल हैं।
Output Tensor Shape: torch.Size([21])
Output Tensor: tensor([232, 129,  57, 474, 576, 577, 106,  24, 355, 578, 232, 579, 580,  10,
        258, 448,  10, 581, 579, 171,   1])
--------------------------------------------------------------------------------------------


Training Batch number-----> 5
Input Language: the yale school of architecture’s mandate is for each student to understand architecture as a creative , productiv
e , innovative , and responsible practice
Input Tensor Shape: torch.Size([26])
Input Tensor: tensor([20,  2,  9, 10, 46, 47, 48, 23, 49, 50, 35, 51, 45, 52, 37, 53, 28, 54,
        28, 55, 28, 14, 56, 57, 18,  1])
Output Language: येल स्कूल ऑफ आर्किटेक्चर का जनादेश प्रत्येक छात्र के लिए एक रचनात्मक , उत्पादक , अभिनव और जिम्मेदार अभ्यास के रूप में वास्तुकला को
समझने के लिए है।
Output Tensor Shape: torch.Size([29])
Output Tensor: tensor([ 2,  5,  6, 52, 53, 54, 55, 56, 14, 32, 57, 58, 24, 59, 24, 60, 10, 61,
        62, 14, 63, 30, 64, 37, 65, 14, 32, 21,  1])
--------------------------------------------------------------------------------------------


Training Batch number-----> 6
```

```
Input Language: yale’s graduate school of arts & sciences offers programs leading to m a  , m s  , m phil  , and ph d  degrees in 
73 departments and programs
Input Tensor Shape: torch.Size([35])
Input Tensor: tensor([33,  8,  9, 10, 11, 12, 13,  3, 26, 34, 35, 36, 37, 18, 28, 36, 38, 18,
        28, 36, 39, 18, 28, 14, 40, 41, 18,  5, 42, 43, 44, 14, 26, 18,  1])
Output Language: येल के ग्रेजुएट स्कूल ऑफ आर्ट्स एंड साइंसेज एमए , एमएस , एम फिल , और पीएचडी के लिए अग्रणी कार्यक्रम प्रदान करता है। 73 विभागों और 
कार्यक्रमों में डिग्री।
Output Tensor Shape: torch.Size([33])
Output Tensor: tensor([ 2, 14,  4,  5,  6,  7,  8,  9, 41, 24, 42, 24, 43, 44, 24, 10, 45, 14,
        32, 46, 47, 19, 20, 21, 48, 49, 10, 23, 30, 50,  1,  0,  0])
-------------------------------------------------------------------------------------------------------------


Training Batch number-----> 7
Input Language: made up of a variety of centers and initiatives that interact with all aspects of the university
Input Tensor Shape: torch.Size([19])
Input Tensor: tensor([208, 209,  10,  37, 191,  10, 192,  14, 201,  88, 210, 211, 194, 212,
         10,  20, 207,  18,   1])
Output Language: विश्वविद्यालय के सभी पहलुओं के साथ बातचीत करने वाले विभिन्न केंद्रों और पहलों से बना है।
Output Tensor Shape: torch.Size([18])
Output Tensor: tensor([232,  14, 216, 234,  14, 235, 236, 117, 199, 218, 220,  10, 222,  16,
        237,  21,   1,   0])
-------------------------------------------------------------------------------------------------------------


Training Batch number-----> 8
Input Language: following are ways for yale graduates and affiliates to stay connected to the university and to the yale network w
orld wide
Input Tensor Shape: torch.Size([23])
Input Tensor: tensor([317, 144, 345,  23,   2,  77,  14, 335,  35, 346, 347,  35,  20, 207,
         14,  35,  20,   2, 226,  82, 190,  18,   1])
Output Language: येल स्नातकों और संबद्धों के लिए विश्वविद्यालय और येल नेटवर्क के विश्वव्यापी नेटवर्क से जुड़े रहने के तरीके निम्नलिखित हैं।
Output Tensor Shape: torch.Size([23])
Output Tensor: tensor([  2,  88,  10, 391,  14,  32, 232,  10,   2, 260,  14, 392, 260,  16,
        393, 394,  14, 395, 362, 171,   1,   0,   0])
-------------------------------------------------------------------------------------------------------------


Training Batch number-----> 9
Input Language: yale is committed to sustainability and has implemented numerous environmental initiatives on campus
Input Tensor Shape: torch.Size([15])
Input Tensor: tensor([  2,  48, 145,  35, 493,  14,  59, 494, 230, 495, 201,  25, 479,  18,
          1])
Output Language: येल स्थिरता के लिए प्रतिबद्ध है और परिसर में कई पर्यावरणीय पहलों को लागू किया है।
Output Tensor Shape: torch.Size([19])
Output Tensor: tensor([  2, 556,  14,  32, 170, 106,  10, 538,  30, 258, 557, 222,  37, 558,
        244,  21,   1,   0,   0])
```

```
-------------------------------------------------------------------------------------------


Training Batch number-----> 10
Input Language: yale center for british art to the peabody museum of natural history and numerous smaller collections , are integr
al parts of teaching and open to the public
Input Tensor Shape: torch.Size([29])
Input Tensor: tensor([  2, 240,  23, 241,  58,  35,  20, 242, 243,  10, 244,  62,  14, 230,
        245, 204,  28, 144, 246, 247,  10, 186,  14, 248,  35,  20, 139,  18,
          1])
Output Language: प्राकृतिक इतिहास के पीबॉडी संग्रहालय के लिए येल सेंटर फॉर ब्रिटिश आर्ट और कई छोटे संग्रह , शिक्षण के अभिन्न अंग हैं और जनता के लिए खुले
हैं।
Output Tensor Shape: torch.Size([29])
Output Tensor: tensor([273,  74,  14, 274, 275,  14,  32,   2, 276, 277, 278,  67,  10, 258,
        279, 280,  24, 213,  14, 281, 282, 283,  10, 284,  14,  32, 285, 171,
          1])
-------------------------------------------------------------------------------------------


Training Batch number-----> 11
Input Language: please contact directly the school to which you are applying for their list of fellowships and financial aid oppor
tunities
Input Tensor Shape: torch.Size([21])
Input Tensor: tensor([310, 311, 276,  20,   9,  35, 254, 312, 144, 286,  23, 223, 313,  10,
        309,  14, 287, 294, 137,  18,   1])
Output Language: कृपया सीधे उस स्कूल से संपर्क करें जिसमें आप फेलोशिप और वित्तीय सहायता के अवसरों की सूची के लिए आवेदन कर रहे हैं।
Output Tensor Shape: torch.Size([24])
Output Tensor: tensor([351, 315, 352,   5,  16, 353, 354, 355, 356, 349,  10, 323, 249,  14,
        240, 129, 357,  14,  32, 317, 251, 358, 171,   1])
-------------------------------------------------------------------------------------------


Training Batch number-----> 12
Input Language: the yale school of engineering & applied science is at the cutting edge of research to develop technologies that a
ddress global societal problems
Input Tensor Shape: torch.Size([25])
Input Tensor: tensor([ 20,   2,   9,  10,  93,  12,  94,  95,  48,  96,  20,  97,  98,  10,
         32,  35,  99, 100,  88, 101, 102, 103, 104,  18,   1])
Output Language: येल स्कूल ऑफ इंजीनियरिंग एंड एप्लाइड साइंस वैश्विक सामाजिक समस्याओं का समाधान करने वाली प्रौद्योगिकियों को विकसित करने के लिए अनुसंधान
के अत्याधुनिक रूप में है।
Output Tensor Shape: torch.Size([27])
Output Tensor: tensor([  2,   5,   6, 109,   8, 110, 112, 113, 114, 115,  53, 116, 117, 118,
        119,  37, 120, 117,  14,  32,  28,  14, 121,  63,  30,  21,   1])
-------------------------------------------------------------------------------------------


Training Batch number-----> 13
```

```
Input Language: search this site to discover the range of yale’s international centers and initiatives , study abroad and exchange
programs , collections , and galleries
Input Tensor Shape: torch.Size([26])
Input Tensor: tensor([196, 197, 198,  35, 199,  20, 200,  10,  33, 164, 192,  14, 201,  28,
         27, 202,  14, 203,  26,  28, 204,  28,  14, 205,  18,   1])
Output Language: येल के अंतरराष्ट्रीय केंद्रों और पहलों की श्रेणी , विदेश में अध्ययन और विनिमय कार्यक्रमों , संग्रहों और दीर्घाओं की खोज के लिए इस साइट को
खोजें।
Output Tensor Shape: torch.Size([28])
Output Tensor: tensor([  2,  14, 195, 220,  10, 222, 129, 223,  24, 224,  30,  22,  10, 225,
         23,  24, 226,  10, 227, 129, 228,  14,  32, 229, 230,  37, 231,   1])
-------------------------------------------------------------------------------------------------------------
```

Test data loader

In [18]:

```python
for batch_index, packed in enumerate(test_loader):
    input_tensors, output_tensors, input_language, out_language = packed
    print("\033[1mTesting Batch number-----> {}\033[0m".format(batch_index+1))
    # print the first input and output tensors along with their respective languages
    print("Input Language:", input_language[0])
    print("Input Tensor Shape:", input_tensors[0].shape)
    print("Input Tensor:", input_tensors[0])

    print("Output Language:", out_language[0])
    print("Output Tensor Shape:", output_tensors[0].shape)
    print("Output Tensor:", output_tensors[0])
    print('-------------------------------------------------------------------------------------------------------------')
    print("\n")
```

```
Testing Batch number-----> 1
Input Language: yale offers significant financial assistance to international students to cover tuition costs as it does with stud
ents from the u s
Input Tensor Shape: torch.Size([23])
Input Tensor: tensor([  2,   3, 302, 287, 220,  35, 164, 142,  35, 303, 304, 305,  52, 306,
        307, 211, 142, 238,  20, 308,  38,  18,   1])
Output Language: येल अंतरराष्ट्रीय छात्रों को ट्यूशन की लागत को कवर करने के लिए महत्वपूर्ण वित्तीय सहायता प्रदान करता है जैसा कि यह यू एस  के छात्रों के साथ
करता है।
Output Tensor Shape: torch.Size([31])
Output Tensor: tensor([  2, 195, 261,  37, 339, 129, 340,  37, 341, 117,  14,  32, 342, 323,
        249,  19,  20, 106, 343, 344, 345, 346, 347, 348,  14, 261,  14, 235,
         20,  21,   1])
-------------------------------------------------------------------------------------------------------------
```

```
Testing Batch number-----> 2
Input Language: opportunities for study or research abroad as well as exchange programs are managed by the individual schools and
programs
Input Tensor Shape: torch.Size([21])
Input Tensor: tensor([137,  23,  27, 213,  32, 202,  52, 214,  52, 203,  26, 144, 215, 216,
         20, 217,  17,  14,  26,  18,   1])
Output Language: विदेशों में अध्ययन या अनुसंधान के अवसरों के साथ-साथ विनिमय कार्यक्रमों का प्रबंधन व्यक्तिगत स्कूलों और कार्यक्रमों द्वारा किया जाता है।
Output Tensor Shape: torch.Size([22])
Output Tensor: tensor([238,  30,  22, 239,  28,  14, 240,  14, 241, 225,  23,  53, 164, 242,
         13,  10,  23, 243, 244, 245,  21,   1])
-------------------------------------------------------------------------------------------
```

Validation data loader

In [19]:
```python
for batch_index, packed in enumerate(val_loader):
    input_tensors, output_tensors, input_language, out_language = packed
    print("\033[1mValidation Batch number-----> {}\033[0m".format(batch_index+1))
    # print the first input and output tensors along with their respective languages
    print("Input Language:", input_language[0])
    print("Input Tensor Shape:", input_tensors[0].shape)
    print("Input Tensor:", input_tensors[0])

    print("Output Language:", out_language[0])
    print("Output Tensor Shape:", output_tensors[0].shape)
    print("Output Tensor:", output_tensors[0])
    print('-------------------------------------------------------------------------------------------')
    print("\n")
```

```
Validation Batch number-----> 1
Input Language: the jackson school of global affairs trains and equips a new generation of leaders to devise thoughtful , evidence
-based solutions for challenging global problems
Input Tensor Shape: torch.Size([26])
Input Tensor: tensor([ 20, 114,   9,  10, 102, 115, 116,  14, 117,  37, 118, 119,  10,  72,
         35, 120, 121,  28, 122, 123,  23, 124, 102, 104,  18,   1])
Output Language: जैक्सन स्कूल ऑफ ग्लोबल अफेयर्स चुनौतीपूर्ण वैश्विक समस्याओं के लिए विचारशील , साक्ष्य-आधारित समाधान तैयार करने के लिए नेताओं की एक नई
पीढ़ी को प्रशिक्षित और सुसज्जित करता है।
Output Tensor Shape: torch.Size([30])
Output Tensor: tensor([135,   5,   6, 136, 138, 139, 113, 115,  14,  32, 140,  24, 141, 116,
        142, 117,  14,  32,  82, 129,  57, 143, 144,  37, 145,  10, 146,  20,
         21,   1])
-------------------------------------------------------------------------------------------
```

```
Validation Batch number-----> 2
Input Language: the institution is also led and supported by the university cabinet
Input Tensor Shape: torch.Size([13])
Input Tensor: tensor([ 20, 449,  48, 314, 450,  14, 451, 216,  20, 207, 452,  18,   1])
Output Language: संस्था का नेतृत्व और समर्थन विश्वविद्यालय मंत्रिमंडल द्वारा भी किया जाता है।
Output Tensor Shape: torch.Size([13])
Output Tensor: tensor([508,  53, 509,  10, 209, 232, 510, 243, 361, 244, 245,  21,   1])
--------------------------------------------------------------------------------------------------------------
```