Neural Machine Translation

Objective:

The field of machine translation (MT), the automatic translation of written text from one natural language into another, has experienced a major paradigm shift in recent years [1]. The objective of this project is to create a Neural machine translation (NMT) on English Telugu parallel dataset [2].

Neural Machine Translation:

Machine Translation (MT) is a subfield of computational linguistics focusses on translating text from one language to another, in this case English to Telugu. With the power of deep learning, Neural Machine Translation (NMT) has arisen as the most powerful algorithm to perform this task. The Recurrent Models are used for sequential data like this. Using a Bidirectional Networks helps in converting the other way around.

Dataset Description:

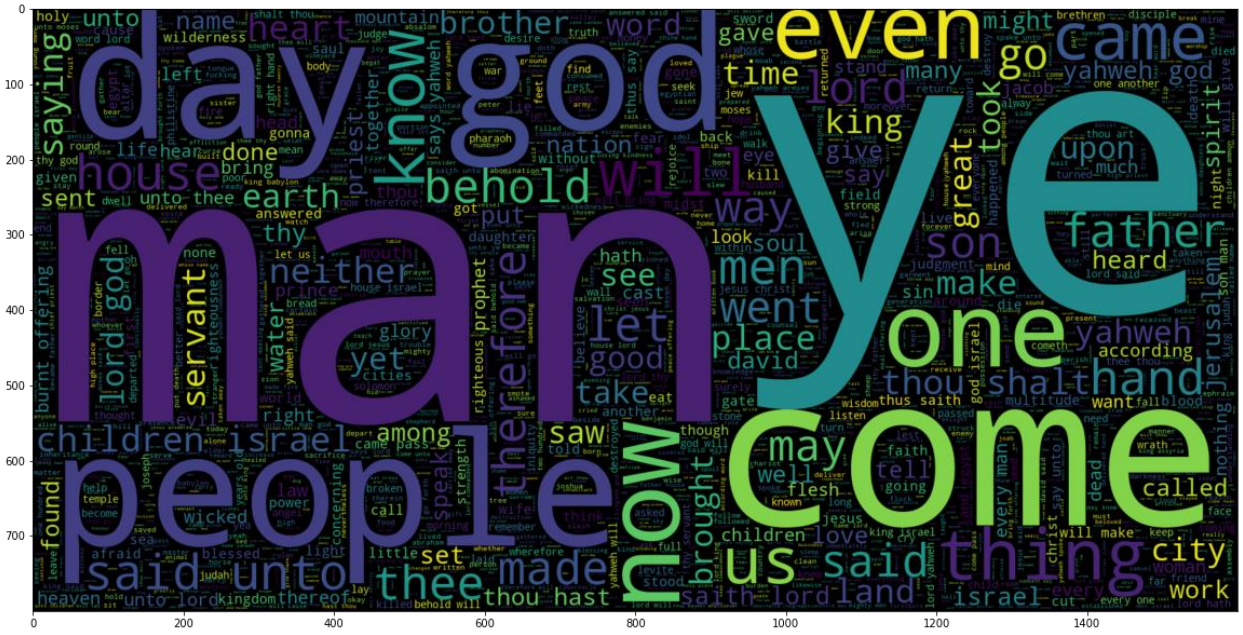
The dataset contains 6 files each training, testing and validation for Telugu and English languages. Training data set contains 75000 sentences. The data has been cleaned from hyperlinks and comments. The data has been already converted to lowercase.

English Data description:

```
The number of tokens is 1809312
The average number of tokens per sentence is 24
The number of unique tokens are 21095
```

Telugu Data description:

```
The number of tokens is 1030924
The average number of tokens per sentence is 14
The number of unique tokens are 106016
```



Progress: [3]

- Reading data

```
f= open("/content/drive/My Drive/English-Telugu/train.en")
en=f.readlines()
len(en)
```

```
75000
```

- Describing Data, with tokenization and unique words

```
from nltk.tokenize import word_tokenize,sent_tokenize
filepath = nltk.data.find('/content/drive/My Drive/English-Telugu/train.te')
corpus = open(filepath, 'r').read()
words = nltk.word_tokenize(corpus)
print("The number of tokens is", len(words))
average_tokens = round(len(words)/75000)
print("The average number of tokens per sentence is",average_tokens)
unique_tokens = set(words)
print("The number of unique tokens are", len(unique_tokens))
```

```
The number of tokens is 1030924
The average number of tokens per sentence is 14
The number of unique tokens are 106016
```

- Cleaned Data set Using following function, which removes punctuation

```
import re
def remove_punc(x):
  return re.sub('[!#?,.:";\n]', '', x)
```

- Tokenizing and padding, for neural network input

```
def tokenize_and_pad(x, maxlen):
  #  a tokenier to tokenize the words and create sequences of tokenized words
  tokenizer = Tokenizer(char_level = False)
  tokenizer.fit_on_texts(x)
  sequences = tokenizer.texts_to_sequences(x)
  padded = pad_sequences(sequences, maxlen = maxlen, padding = 'post')
  return tokenizer, sequences, padded
```

- Results after padding

```
[print("The tokenized version for document\n", en[-1:][0],"\n", x_padded[-1:][0])]
```

```
The tokenized version for document
 how could you stand for it
 [149 391   7 316  11  16   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0]
```

```
print("The tokenized version for document\n", te[-1:][0],"\n ", y_padded[-1:][0])
```

```
The tokenized version for document
 మీరు ఎలా నిలబడటానికి కాలేదు
  [   6  296 8102  887    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0]
```

Model1:

```
NMT1=models.load_model('/content/drive/My Drive/NMT_models/NMT1.h5')
NMT1.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 74, 256)           5810688
_________________________________________________________________
lstm_2 (LSTM)                (None, 256)               525312
_________________________________________________________________
repeat_vector_1 (RepeatVecto (None, 47, 256)           0
_________________________________________________________________
lstm_3 (LSTM)                (None, 47, 256)           525312
_________________________________________________________________
time_distributed_1 (TimeDist (None, 47, 106518)        27375126
=================================================================
Total params: 34,236,438
Trainable params: 34,236,438
Non-trainable params: 0
_________________________________________________________________
```

Bleu score:

```
BLEUscore1 = nltk.translate.bleu_score.corpus_bleu(yp1, yt)
print(BLEUscore1)
```

```
0.4779748973490351
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
```

Model2:

```
NMT2=models.load_model('/content/drive/My Drive/NMT_models/NMT2.h5')
NMT2.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 74, 256)           5810688
_________________________________________________________________
lstm (LSTM)                  (None, 74, 256)           525312
_________________________________________________________________
lstm_1 (LSTM)                (None, 128)               197120
_________________________________________________________________
repeat_vector (RepeatVector) (None, 47, 128)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 47, 256)           394240
_________________________________________________________________
lstm_3 (LSTM)                (None, 47, 128)           197120
_________________________________________________________________
time_distributed (TimeDistri (None, 47, 106518)        13740822
=================================================================
Total params: 20,865,302
Trainable params: 20,865,302
Non-trainable params: 0
```

Bleu score:

```
BLEUscore2 = nltk.translate.bleu_score.corpus_bleu(yp2, yt)
print(BLEUscore2)
```

```
0.48104068819490353
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
```

Results[4]:

```
[42] for i in range(1000,1006):
       print("Original Telugu {}\n".format(pad_to_text(y_test[i], y_tokenizer)))
       print("NMT1 Telugu {}\n".format(predictionNMT1(x_test[i:i+1])))
       print("NMT2 Telugu {}\n\n".format(predictionNMT2(x_test[i:i+1])))
```

```
Original Telugu న్యాయమునుబట్టి అయన తీర్పు తీర్చునుఅయన ప్రతిదినము కోపపడు దేవుడు

NMT1 Telugu అందుకు అయన అయన అయన అయన

NMT2 Telugu నీవు మీరు నీవు అయన అయన అయన
```

Original Telugu ఆయన పరలోకమునకు వెళ్లి దూతలమీదను అధికారుల మీదను శక్తులమీదను అధికారము పొందినవాడై దేవుని కుడిపార్శ్వమున ఉన్నాడు

NMT1 Telugu ఆయన ఆ ఆ ఆయన ఆయన ఆయన ఆయన అది అది

NMT2 Telugu వారు తన తన తన తన తన తన తన తన

Original Telugu మరియు మీలో ఎవడు చింతిచుటవలన తన యెత్తును మూరెడెక్కువ చేసికొన గలడు

NMT1 Telugu ఆయన తన తన ఆయన అది అది

NMT2 Telugu ఆయన తన ఆయన ఆయన ఆయన

The NMT2 model has better accuracy than NMT1

References:

[1] Neural Machine Translation: A Review and Survey (https://arxiv.org/abs/1912.02047)

[2] https://github.com/himanshudce/Indian-Language-Dataset

[3] https://colab.research.google.com/drive/1vrF7vr8M4LOpPAdj2vgZVwU5Ltkkgwwh?usp=sharing

[4] https://colab.research.google.com/drive/10VbTKC7ajIIpemepl78ENJeQ9FXlSz1-?usp=sharing