# **Machine Translation**

The task is to train a machine translation model, which takes natural language instructions and converts them into a machine readable format. The data is a sample from survey questionnaires, where each question is asked based on a specific answer selected in some previous question.

### **Training Data:** MT\_training\_corpus.xlsx

1. QID – contains question ID of the current question and the question ID on which the current question is dependent.
2. CONDITION – contains the natural language condition statement.
3. OUTPUT – contains the target machine readable format to be generated.

### **Test Data:** MT\_test\_submission.xlsx

1. QID – contains question ID of the current question and the question ID on which the current question is dependent.
2. CONDITION – contains the natural language condition statement.
3. OUTPUT – Must be predicted by your trained model.

**Note: Do NOT shuffle the rows of test file, as they will be evaluated by a machine.**

Designed Neural Model for Machine Translation

1. Model overview:

Recurrent Neural Networks (RNNs) are used to create a sequnce to sequnce (seq2seq) model.

A Recurrent Neura Networks with Long Short Term Memory (LSTM) a layer is used to effciently create a laten represnation of input sequnec. This first part of the seq2seq model is refred to as Encoder part of the model.

The latent represnation of the model (hidden states) is fed to another Recurrent Neura Network with LSTM cells followwd by a fully connected layer and a time distributed fully conncted layer. Finally a softmax activation function is used to predcit the diffrenr sequnce words probabities of each time step. This second part of Seq2Seq model is refred to as Decoder part of the model.

To train the network the input sequnce is fed to Enocder. The encoder hidden state and the target sequnce (prepadded by start of sequnce token) is fed to the decoder. The loss function id defined such that Decoder should predcit the targer sequnce step ahead of being fead to Decoder.

During predcition, the encoder hidden state and a start of sequnce token is fed to decoder, The first predcited sequnce by decoder is fed back to itself in next step. This procedure will contunite untill a special token (end of sequnce token) is predcited or maximum sequnce length is reached.

To increase the prediction accuracy of the model a beam search algorithms is used.

2. Training The Model

The main steps to train a seq2seq model:

1. Read dataset

2. Preproces each sequnce (create standarized sequnces)

a. Change QID, CONDITION and OUTPUT text to lowercase

b. split QID, CONDITION and OUTPUT text into tokens (words)

c. Replace QID tokens in each sample with standrized tokens (i.e., <QID0>, <QID1>, ...)

d. Replace digit tokens in each sample with standarized tokens (i.e., <DGT0>, <DGT1>, ...)

e. Create standardization dictionary for each sample

f. Add special tokens <BOS> and <EOS> to the begining and end of each sequence

3. Create dictinries to convert input and target sequnces to an integer id

4. Replace input and outpu sequnce tokens with an integre id

5. Pad sequnces with zero to create fixed size input and target sequnces

a. Input sequnce is pre-padded with zero

b. Target sequnce is post-padded

4. Create a seq2seq model

4. Train the model

5. Save the model and model metadata (inclding dictionaries to conver words to id)

Model can be trained based on Training Data (MT\_training\_corpus.xlsx) by running the attached python script (train.py). The required packages are included in “requirement.txt”

$ pip3 install requirements.txt

$ python3 train.py

Model can also be trained by running the train file cell in the include Jupter notebook (“machine\_translation.ipynb”)

3. Prediction Using the Model

The main steps to predict an output sequnce using the seq2seq model:

1. Read test dataset

2. Preproces each sequnce (create standarized sequnces)

a. Change QID and CONDITION text to lowercase

b. split QID and CONDITION text into tokens (words)

c. Replace QID tokens in each sample with standrized tokens (i.e., <QID0>, <QID1>, ...)

d. Replace digit tokens in each sample with standarized tokens (i.e., <DGT0>, <DGT1>, ...)

e. Create standardization dictionary for each sample

f. Add special tokens <BOS> and <EOS> to the begining and end of each sequence

3. Replace condition sequnce tokens with an integre id usng the encoder\_word2id dictionary

4. Pad condition sequnce with zero to create a fixed size input sequnce

a. Input sequnce is pre-padded with zero

5. Extract Encoder and Decoder parts of saved seq2seq model

6. Use a beam search algorithm to predict the output sequnce

7. Reverse predicted output sequnce to words using the decoder\_id2word dictionary

8. Revrese Digit and QID standardization from the predicted output

9. Save the precited outputs to a file

Once a model is trained, it can be used for predcition of the provided Test Data (MT\_test\_submission.xlsx) by running the attached python script (predict.py).

$ python3 predict.py

Test data can also be predcited by running the predict file cell in the include Jupter notebook (“machine\_translation.ipynb”). The outputs will be saved in “./data/ MT\_test\_submission\_with\_predcitions.xlsx”.

4. Model Performance

The trained model provides accuracy of ~95% on the validation set which is remarakble considering that train dataset is not large.

The following some of the model predcitions on training data.

Sample index 0

QID: Q26,Q20

CONDITION: ASK Q26 IF Q20 = 0

OUTPUT: Q20.any(0)

Predicted OUTPUT: q20.any(0)

Sample index 10

QID: 8012,8042,8050

CONDITION: IF 8012= 1,2,3 or 4 AND 8042=1,2 or 3 AND 8050=1 then classify as subgroup A

OUTPUT: 8012.any(1,2,3,4)&8042.any(1,2,3)&8050.any(1)

Predicted OUTPUT: 8012.any(1,2,4,3,4,

Sample index 20

QID: QE2,hQK

CONDITION: IF QE2 = NOT 99, CONTINUE, OTHERWISE END AT hQK

OUTPUT: QE2.notany(99)

Predicted OUTPUT: qe2.notany(99)

Sample index 30

QID: Q18A,Q18D,QD

CONDITION: PROGRAMMER: Q18A-Q18D ARE ASKED ONLY IF QD = 2 (BBQ RANCH CHICKEN SALAD)

OUTPUT: QD.any(2)

Predicted OUTPUT: qd.any(2)

Model predictions on some test data are as follows.

Sample index 0

QID: 1010

CONDITION: Terminate if respondent selected ‘A4’ for all 3 product types

OUTPUT: nan

Predicted OUTPUT: 1010.any(3)

Sample index 10

QID: Q16A,QD

CONDITION: ASK ONLY IF QD = 1-4 (ORDERED ANY TEST SALAD)

OUTPUT: nan

Predicted OUTPUT: qd.between(1:4)

Sample index 20

QID: QR8,Q30,QR7

CONDITION: ASK IF Q30=1 AND QR7=1-4

OUTPUT: nan

Predicted OUTPUT: q30.any(1)&qr7.any(4)

Beam search is used to make a better prediction on test data, the model makes several prediction and chooses the best possible sequnce. However this slowes down predcition. For faster predcition number of beam search branches should be set to 1 (not use beam search).

The model performance can be enhanced by:

1. Training on a larger train data and increasing model size

2. Further optimzation of the model paramet

3. Using Embedding layer at Encoder and Decoder inputs

3. Using Attaention Method

The seq2seq model is implemnted uing Tensorflow and is trained based on