PROJECT PROGRESS REPORT (B-TECH CSE IV YEAR)

TITLE OF PROJECT	AI Chatbot (Web Assistant)
GROUP MEMBERS	1. Pritam Banik - (1683910031) 2. Sarvesh Kumar - (1683910039) 3. Suman Saurabh - (1683910045) 4. Vishnukant Mishra - (1683910051)
OBJECTIVE OF PROJECT WORK	● To Design a Deep Neural Network and Deep NLP Based Conversational Bot. ● To design a Model which can perform better than existing models. ● To increase the user interactivity in websites and make any official website more user friendly. ● To design a chatbot which allows businesses to meet the demand of Speed and Convenience by giving an immediate response to queries or issues. ● To design an advanced chatbot can remember customer preferences and provide advice, tips and help, while gently upselling in Business Dealing. ● A chatbot which can Deliver the fast and frictionless experience to the customers demand, and they will repay us with loyalty. A 2% increase in customer retention has the same effect on the bottom line as decreasing costs by 10%.
PLATFORM/TECHNOLOGY DETAILS	**Python:** 3.5.0 **Tensorflow:** 1.0.0 (Training on Google Colaboratory) **Dataset:** Cornell movie dialog corpus (Approx 2 Lacs conversation dialogue)

MODULES/PARTS COMPLETED TILL DATE	**Module - 1: Data Preprocessing (NLP)**

	❖ **Data gathering**
	    ➢ Creation of training dataset
	    ➢ Creation of test dataset
	❖ **Data Cleaning**
	    ➢ Removal of stopwords
	    ➢ Taking care of bigrams and polygrams
	    ➢ Stemming of data
	❖ **Making Key-Target Pairs(Question-Answer form)**
	    ➢ Proper label encoding each key-target values
	
	Module - 2: Model Architecture (Seq2Seq Model)
	
	● Creation of RNN/LSTM layer
	● Encoder layer
	● Decoder layer
	
	Module - 3: Training
	
	● Deciding the hyper-parameters
	● Deciding the embedding size
	● Deciding the number of rnn-cells
	● Deciding the learning rate
	● Deciding the batch-size
	● Making Decoder Attention based, so that it can produce the output as a normal human being.

REMAINING MODULES	**Left-Over Task:** Model is trained But have to trained more for better Accuracy and Performance. **Module - 4:** Testing of the Model on unknown Dataset **Module - 5:** Deployment on the Cloud based platform

Screenshots of Completed Module

1. Data preprocessing Module:

Extracted Conversation Lines From Dataset:

nde ▲	Type	Size	Value
0	str	1	Well, I thought we'd start with pronunciation, if that's okay with you ...
1	str	1	Not the hacking and gagging and spitting part. Please.
2	str	1	Okay... then how 'bout we try out some French cuisine. Saturday? Nig ...
3	str	1	Forget it.
4	str	1	Cameron.
5	str	1	The thing is, Cameron -- I'm at the mercy of a particularly hideous br ...
6	str	1	Seems like she could get a date easy enough...
7	str	1	Unsolved mystery. She used to be really popular when she started high ...
8	str	1	That's a shame.

Processed and Cleaned Answers:

nde ▲	Type	Size	Value
0	str	1	well i thought we would start with pronunciation if that is okay with ...
1	str	1	not the hacking and gagging and spitting part please <EOS>
2	str	1	okay then how 'bout we try out some french cuisine saturday night <E ...
3	str	1	forget it <EOS>
4	str	1	cameron <EOS>
5	str	1	the thing is cameron I am at the mercy of a particularly hideous bree ...
6	str	1	seems like she could get a date easy enough <EOS>
7	str	1	unsolved mystery she used to be really popular when she started high ...
8	str	1	that is a shame <EOS>

Encoding Lines into Integer for input to Encoder Layer:

ıde ▲	Type	Size	Value
0	list	26	[15, 48, 25, 47, 49, 50, 51, 15, 52, 53, ...]
1	list	26	[8826, 63, 60, 64, 65, 66, 67, 68, 69, 60, ...]
2	list	2	[102, 8825]
3	list	8	[1531, 77, 101, 1552, 33, 149, 608, 8825]
4	list	6	[27, 153, 227, 3, 6453, 8825]
5	list	11	[26, 27, 7, 160, 253, 65, 1281, 97, 65, 613, ...]
6	list	3	[1389, 134, 8825]
7	list	7	[27, 239, 133, 194, 226, 74, 8825]
8	list	2	[196, 8825]

2. Training The Model:

```
for epoch in range(4, epochs + 1):
    for batch_index, (padded_questions_in_batch, padded_answers_in_batch) in enumerate(split_into_batches(training_questions, training_answers,
        starting_time = time.time()
        _, batch_training_loss_error = session.run([optimizer_gradient_clipping, loss_error], {inputs: padded_questions_in_batch,
                                                                                                targets: padded_answers_in_batch,
                                                                                                lr: learning_rate,
                                                                                                sequence_length: padded_answers_in_batch.shape[1
                                                                                                keep_prob: keep_probability})

        total_training_loss_error += batch_training_loss_error
        ending_time = time.time()
        batch_time = ending_time - starting_time
        if batch_index % batch_index_check_training_loss == 0:
            print('Epoch: {:>3}/{}, Batch: {:>4}/{}, Training Loss Error: {:>6.3f}, Training Time on 100 Batches: {:d} seconds'.format(epoch,
                                                                                                                                        epochs,
                                                                                                                                        batch_in
                                                                                                                                        len(trai
                                                                                                                                        total_tr
                                                                                                                                        int(batc

            total_training_loss_error = 0
        if batch_index % batch_index_check_validation_loss == 0 and batch_index > 0:
            total_validation_loss_error = 0
            starting_time = time.time()
            for batch_index_validation, (padded_questions_in_batch, padded_answers_in_batch) in enumerate(split_into_batches(validation_questio
                batch_validation_loss_error = session.run(loss_error, {inputs: padded_questions_in_batch,
                                                                        targets: padded_answers_in_batch,
                                                                        lr: learning_rate,
                                                                        sequence_length: padded_answers_in_batch.shape[1
                                                                        keep_prob: 1})
```

```
Validation Loss Error:  1.143, Batch Validation Time: 259 seconds
I speak better now!!
Epoch:   5/100, Batch:     0/5417, Training Loss Error:  0.200, Training Time on 100 Batches: 93 seconds
Epoch:   5/100, Batch:   100/5417, Training Loss Error:  1.161, Training Time on 100 Batches: 92 seconds
Epoch:   5/100, Batch:   200/5417, Training Loss Error:  1.167, Training Time on 100 Batches: 61 seconds
Epoch:   5/100, Batch:   300/5417, Training Loss Error:  1.146, Training Time on 100 Batches: 44 seconds
Epoch:   5/100, Batch:   400/5417, Training Loss Error:  1.154, Training Time on 100 Batches: 31 seconds
Epoch:   5/100, Batch:   500/5417, Training Loss Error:  1.123, Training Time on 100 Batches: 47 seconds
Epoch:   5/100, Batch:   600/5417, Training Loss Error:  1.163, Training Time on 100 Batches: 52 seconds
Epoch:   5/100, Batch:   700/5417, Training Loss Error:  1.179, Training Time on 100 Batches: 36 seconds
Epoch:   5/100, Batch:   800/5417, Training Loss Error:  1.166, Training Time on 100 Batches: 88 seconds
Epoch:   5/100, Batch:   900/5417, Training Loss Error:  1.185, Training Time on 100 Batches: 107 seconds
Epoch:   5/100, Batch:  1000/5417, Training Loss Error:  1.142, Training Time on 100 Batches: 47 seconds
Epoch:   5/100, Batch:  1100/5417, Training Loss Error:  1.171, Training Time on 100 Batches: 174 seconds
Epoch:   5/100, Batch:  1200/5417, Training Loss Error:  1.103, Training Time on 100 Batches: 180 seconds
Epoch:   5/100, Batch:  1300/5417, Training Loss Error:  1.104, Training Time on 100 Batches: 159 seconds
Epoch:   5/100, Batch:  1400/5417, Training Loss Error:  1.186, Training Time on 100 Batches: 74 seconds
Epoch:   5/100, Batch:  1500/5417, Training Loss Error:  1.194, Training Time on 100 Batches: 80 seconds
Epoch:   5/100, Batch:  1600/5417, Training Loss Error:  1.177, Training Time on 100 Batches: 80 seconds
Epoch:   5/100, Batch:  1700/5417, Training Loss Error:  1.164, Training Time on 100 Batches: 72 seconds
Epoch:   5/100, Batch:  1800/5417, Training Loss Error:  1.173, Training Time on 100 Batches: 67 seconds
Epoch:   5/100, Batch:  1900/5417, Training Loss Error:  1.149, Training Time on 100 Batches: 92 seconds
Epoch:   5/100, Batch:  2000/5417, Training Loss Error:  1.207, Training Time on 100 Batches: 47 seconds
Epoch:   5/100, Batch:  2100/5417, Training Loss Error:  1.156, Training Time on 100 Batches: 51 seconds
Epoch:   5/100, Batch:  2200/5417, Training Loss Error:  1.177, Training Time on 100 Batches: 99 seconds
Epoch:   5/100, Batch:  2300/5417, Training Loss Error:  1.207, Training Time on 100 Batches: 149 seconds
Epoch:   5/100, Batch:  2400/5417, Training Loss Error:  1.163, Training Time on 100 Batches: 101 seconds
Epoch:   5/100, Batch:  2500/5417, Training Loss Error:  1.169, Training Time on 100 Batches: 104 seconds
Epoch:   5/100, Batch:  2600/5417, Training Loss Error:  1.184, Training Time on 100 Batches: 124 seconds
Epoch:   5/100, Batch:  2700/5417, Training Loss Error:  1.202, Training Time on 100 Batches: 80 seconds
Validation Loss Error:  1.138, Batch Validation Time: 259 seconds
I speak better now!!
```