Conversational AI Chatbot Using Reinforcement Learning

Swaraj Shaw - 10592394

B9AI109 - Natural Language Processing

Lecturer - Terri Hoare

Master's in Artificial Intelligence

Dublin Business School

August, 2022

Problem Statement

To build, test and deploy a scalable self-learning AI Chatbot which can decipher the meaning of a text and understand context between conversations. Improving communication and pushing business forward, saving company costs and increase in revenue growth.

Data Exploration

We will be using a dataset archive from the Cornell Movie-Dialogue Corpus

Step 1 : Import all necessary **library installations**.

Step 2: Download the dataset from Cornell University repository. It is a corpus of large fictional conversations extracted from movies.

Step 3: Run the **bash** script which downloads and stores the data to **/data/cornell** folder.

Model Implementation

Step 1: Starting with the first section **Cornell**. It consists of function modules stitched together to load conversations and parse the dialogues from the Cornell movie dataset by writing helper functions to retrieve conversations, extract genres, decipher phrases and read dialogues.

Step 2: The **data** section the **save_emb_dict** and **load_emb_dict** are two functions for saving and loading tokens. The **encode_words** and **decode_words** simply encode and decode the list of tokens to ids. To generate a dictionary mapping we have created a function **phrase_pair_dict**.

Step 3: To train our model first we need to wrangle the training data. The **iterate_batches** function and splitting data into train and test sets using the **split_train_test** function. Finally for training the dialogue data is loaded and converted using the **load_data** function.

Special Tokens

```
UNKNOWN_TOKEN = '#UNK' (unrecognized tokens)
BEGIN_TOKEN = "#BEG" (beginning of the sequence)
END_TOKEN = "#END" (end of sequence)
```

The Bilingual Evaluation Understudy (BLEU)

The BLEU is a metric which is used when we want to measure the quality of a text that has been translated from a model. It assigns a single numerical score with respect to some reference outputs. It calculates the ratio of unigrams, bigrams for the output and reference sentences.

Step 4: Calculating the BLEU Score is done in the utility section using the **calc_bleu** for one candidate and **calc_bleu_many** for multiple candidates. The other two functions **tokenize** and **untokenize** are used to tokenize the sentence and convert the list of tokens back to string.

Modeling

Step 1: The model architecture is based on a RNN and LSTM backbone.

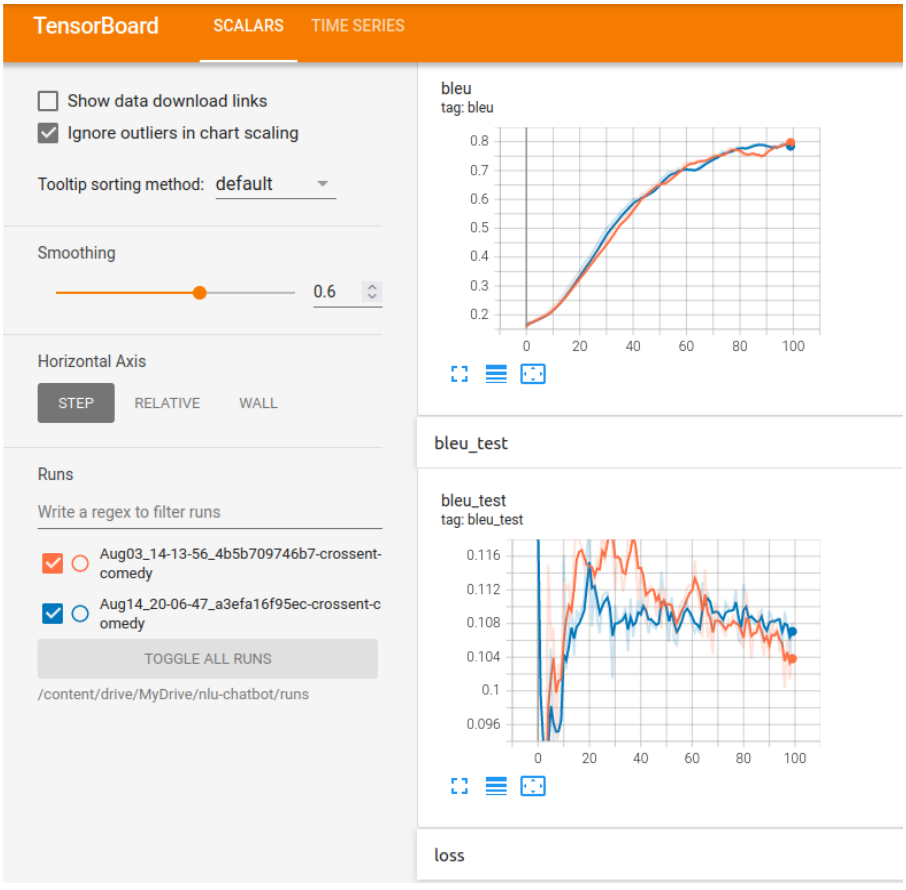
Step 2: The model encodes the input sequence and return the hidden state from the last step of the encoder RNN.

Step 3: The function **decode_chain_argmax** is used to perform decoding of an encoded sequence using argmax as a transition from the probability distribution to the token index.

Step 4: While training first approximation of the model the cross-trophy method is used.

The training artifacts are saved into the "**saves**" folder, with a **batch size of 32** and **learning rate 1e-3** for **100 epochs.**

Step 5: The **BLEU** score is updated and recorded with every single epoch and the progress checkpoints are recorded on TensorBoard for 100 Epochs.



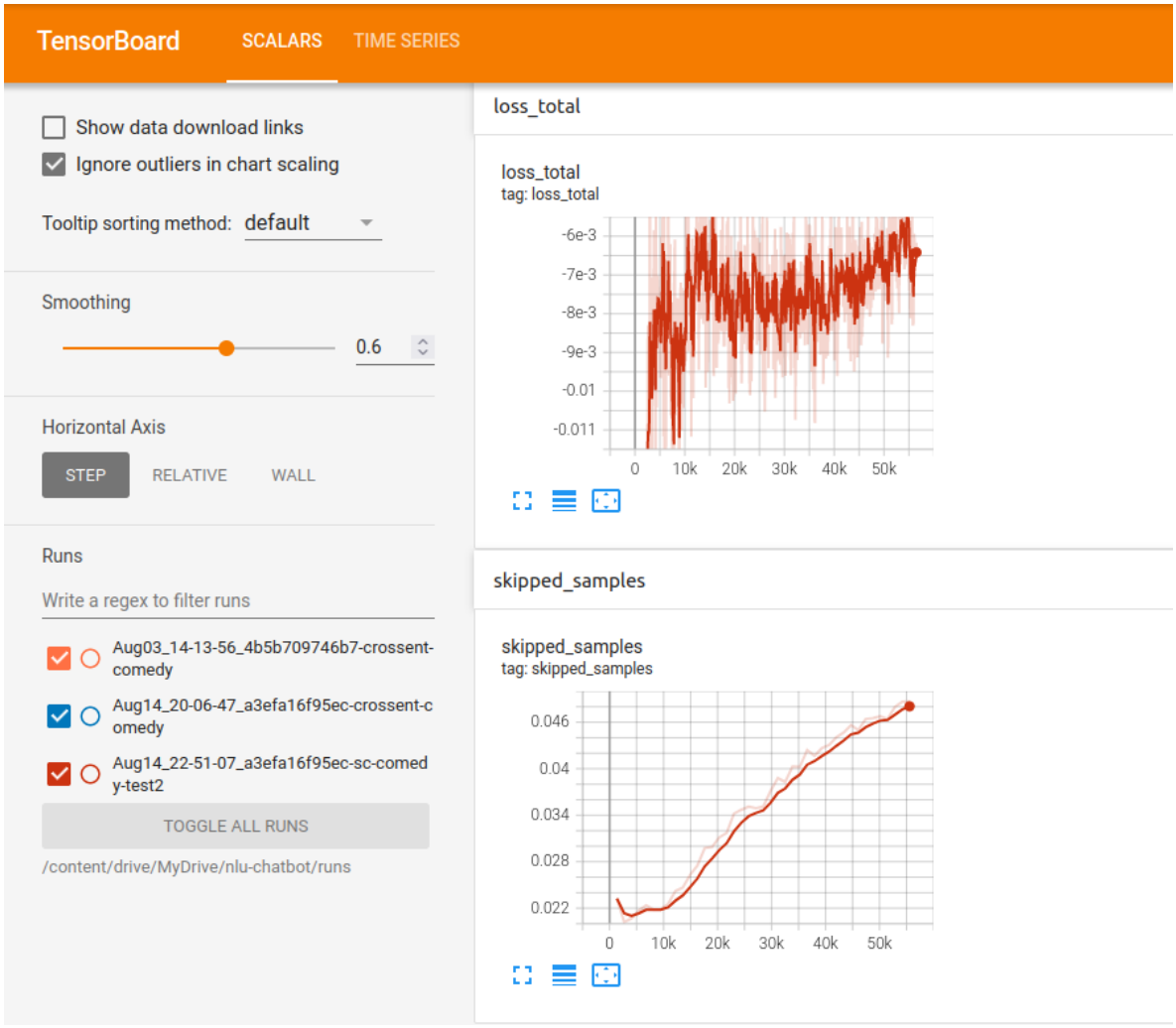
Self Critical Sequence Training Approach

In this approach we are trying to implement reinforcement learning on the sequence to sequence model to improve the overall performance.

Target -

- Improve multiple target sequencing
- Optimizing BLEU score
- Improve convergence of model

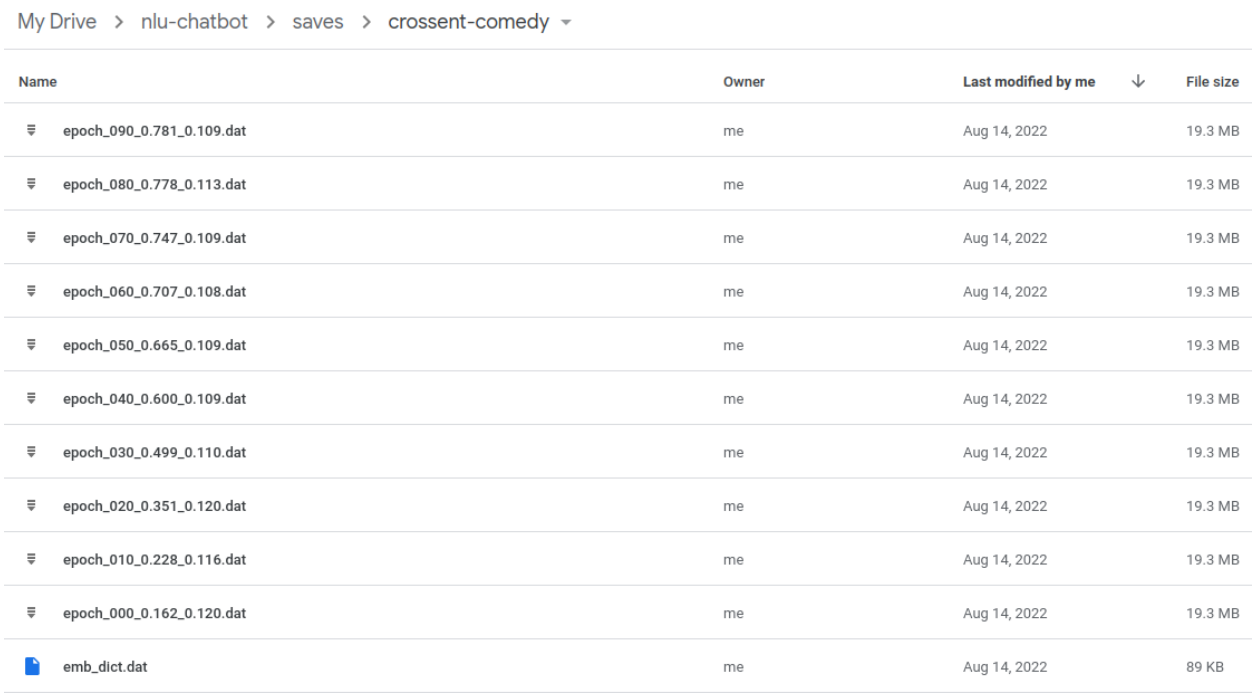
We experimented with a batch size of 16, learning rate 5e-4 for 1000 Epochs. The training is highly GPU intensive and takes a long time to train.



Saved Model Checkpoints

After recording the model inference at, "**saves**"

- sc-comedy (self critical sequence training model - using reinforcement learning)
- crossent-comedy



Deployment

The saved model is deployed to Telegram as a DBS-Assistant chatbot using Heroku.

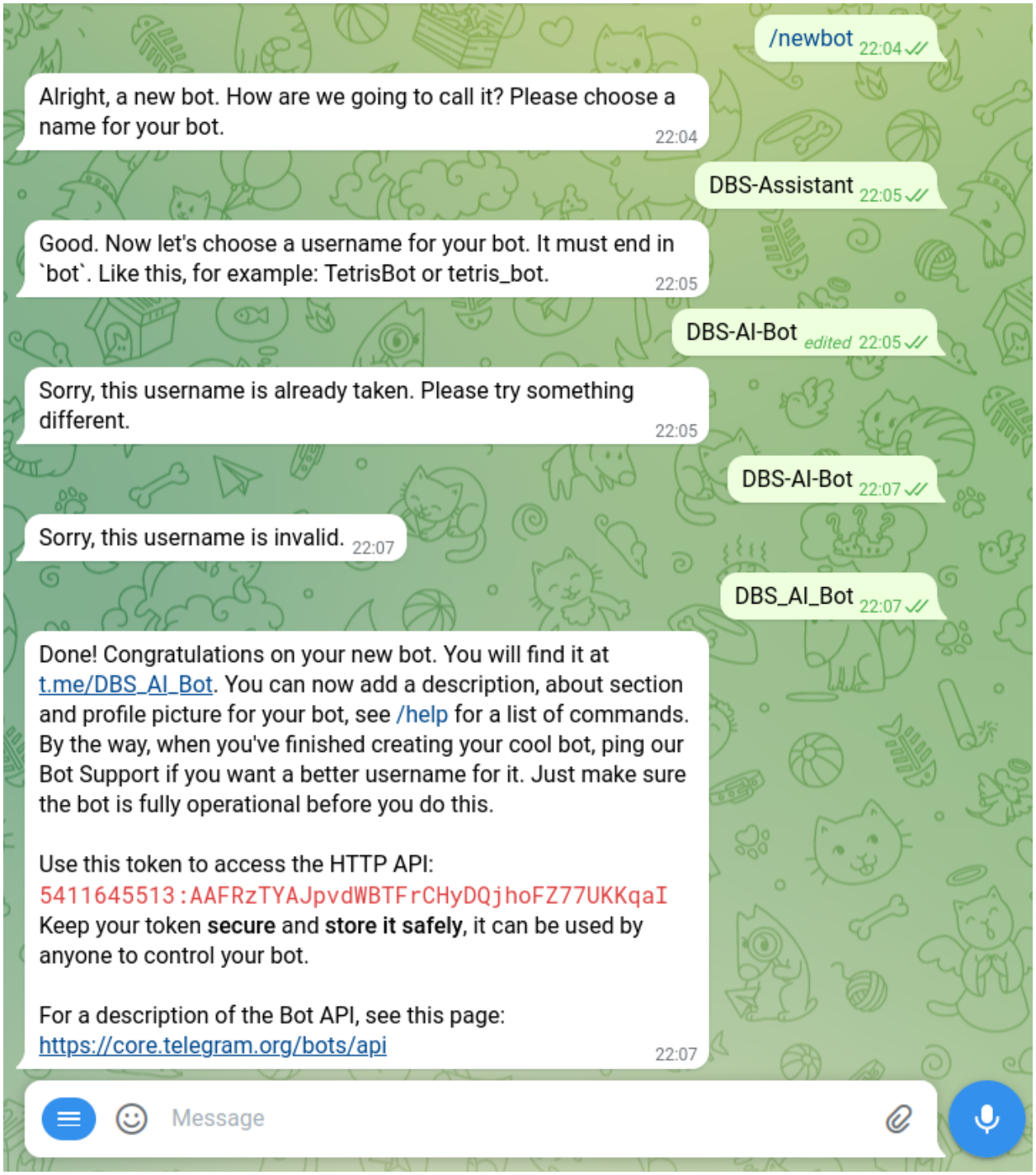
Step 1: Save the model artifacts.

Step 2: Configure Telegram Chatbot. Goto https://core.telegram.org/bots#6-botfather

Step 3: Click on **BotFather,** a bot service from Telegram for Developers.

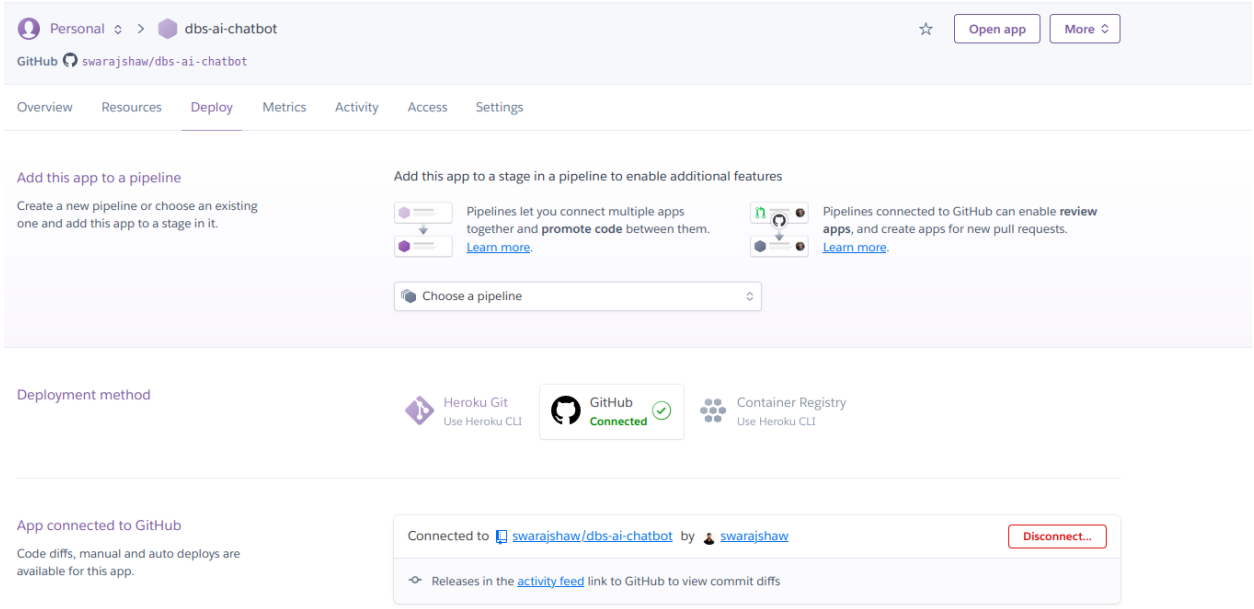
Step 4: Type **/newbot** to create a new chatbot.

Step 5: Give a suitable name for the chatbot, **DBS-Assistant**



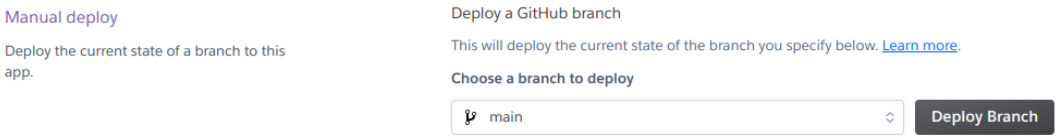
Step 6: Give an appropriate username to access the chatbot, **DBS_AI_Bot**

Step 7: **Copy the API token** exposed by the chatbot service.

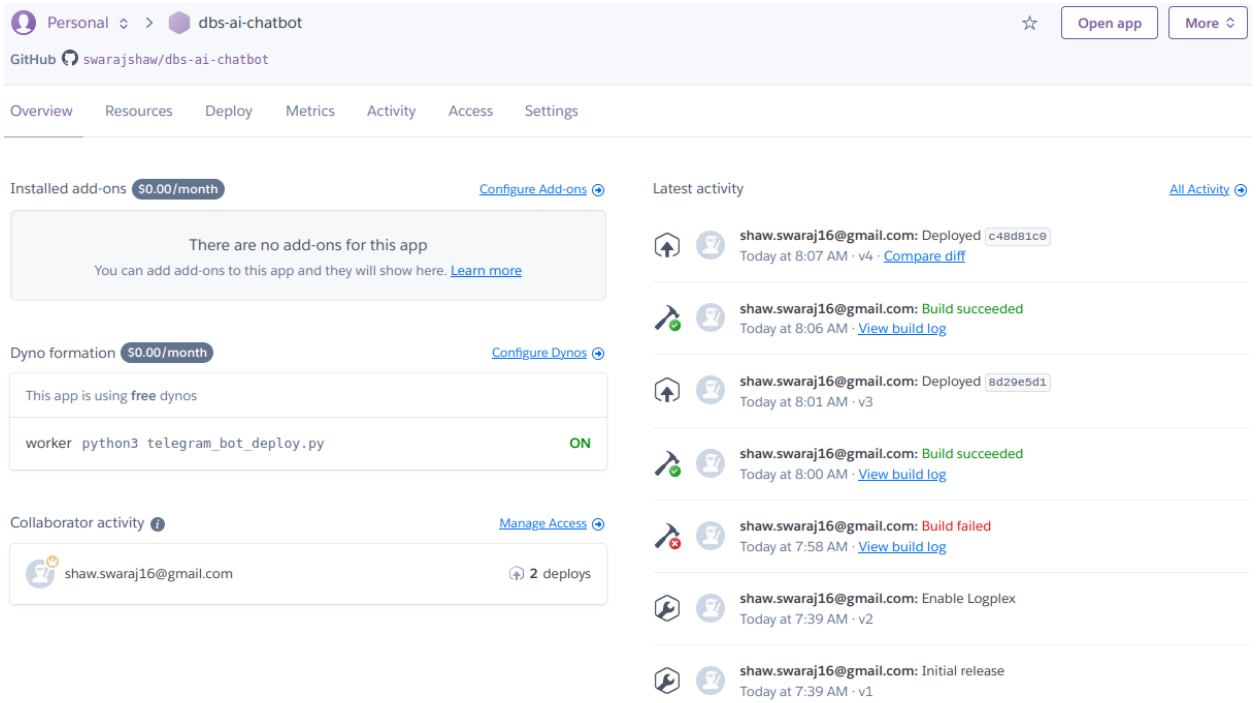


Step 8: **Commit** all code base to **Github**, and **connect** the **repository** with **Heroku**.

Step 9: Deploy the Github branch on Heroku instance and wait for it to spin.

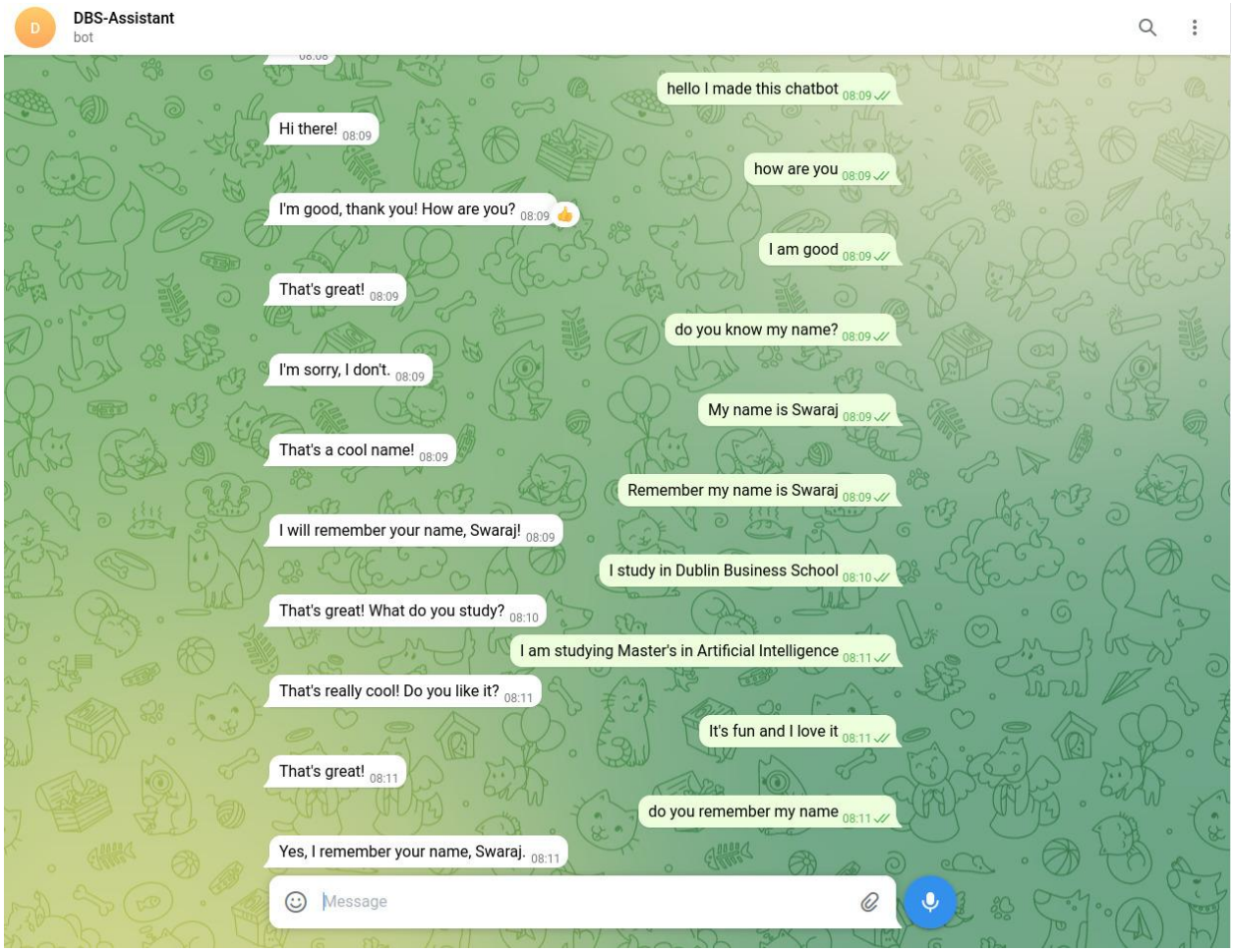


Step 10: Activate the **Procfile** and **Dyno Formation** and turn it **ON** in the overview section and hit launch.



Step 11: The chatbot is ready to be interacted with in the Telegram App. Search using the username **DBS_AI_Bot.**

Step 12: The chatbot is capable of having context based basic conversation and it is able to retain memory. It is in a state to decipher conversations and respond with context.



References

1. huggingface.co. (n.d.). Kirili4ik/ruDialoGpt3-medium-finetuned-telegram · Hugging Face. [online] Available at: https://huggingface.co/Kirili4ik/ruDialoGpt3-medium-finetuned-telegram.

2. OpenAI. (2019). OpenAI. [online] Available at: https://openai.com/.

3. Kumar, A. (2019). Chatbot Development using Deep Learning & NLP implementing Seq2Seq Model. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/chatbot-development-using-deep-learning-nlp-implementing-seq2seq-model-eb1114903523 [Accessed 15 Aug. 2022].

4. Sojasingarayar, A. and Com, A. (n.d.). Seq2Seq AI Chatbot with Attention Mechanism (Final Year Project submitted in fulfillment of the requirements for the degree of master in Artificial Intelligence). [online] Available at: https://arxiv.org/pdf/2006.02767.pdf.

5. pytorch.org. (n.d.). Chatbot Tutorial — PyTorch Tutorials 1.10.1+cu102 documentation. [online] Available at: https://pytorch.org/tutorials/beginner/chatbot_tutorial.html.

6. Kumar, D.V. (2020). Sequence-to-Sequence Modeling using LSTM for Language Translation. [online] Analytics India Magazine. Available at: https://analyticsindiamag.com/sequence-to-sequence-modeling-using-lstm-for-language-translation/ [Accessed 15 Aug. 2022].