Individual Report on Chatbot

STUDENT NAME: SHEHRYAR SHEHRYAR

STUDENT ID: P2952028

MODULE TITLE: NATURAL LANGUAGE PROCESSING

WORD COUNT: 803

Shehryar Shehryar
DE MONTFORT UNIVERSITY, LEICESTER

In this coursework, I construct a chatbot using the Cornell Movie-Dialogs Corpus and sequence-to-sequence models. My key focus was to build and compare multiple neural architectures and choose the best model for interactive-chat, all the way from training data to deployed model.

Data Acquisition and Preprocessing:

The first stage of project is to downloading and unzipping the Cornell Movie-Dialogs Corpus. The transfer pair files in data available in the dataset are movie_lines. txt and movie_conversations. txt, which are files that store lines and conversations.

After successfully extracting the data, each line in a dialogue was cleaned. This included lower-casing all text, removing non-alphanumeric elements and trimming the whitespaces. Long dialogues with more than 20 tokens were filtered to simplify the existing training task during training and enhance model convergence. These preprocessing steps enabled making the training data consistent, cleaned and ready for deep learning models.

Vocabulary and Encoding:

A model was cleaned and filtered to generate a vocabulary of 20,000 most frequent words, Special tokens, and unknown words were added. All the sentences were encoded in token and padded as maximum of 20 tokens. These tokenized sequences can be conveniently fed into neural network between a central PyTorch Dataset.

Models Implemented:

In this coursework I applied and compared 5 different models:

1. **MLP (Multi-Layer Perceptron):** A fully-connected feed-forward network that take input embeddings to generate output sequences.
2. **RNN Seq2Seq:** It encodes and decodes input and output sequences.
3. **GRU Seq2Seq:** GRUs are better able to maintain long-term context, as they solve the vanishing gradient problems more effectively, which explains why their sequence modeling performance is ultimately superior.
4. **LSTM Seq2Seq:** Utilizes Long Short-Term Memory units in the encoder-decoder. Long Short-Term Memory (LSTM) block help to keep track of long-range context, and hence the model can generate more coherent and applicable replies across items for longer history length.
5. **Transformer:** A reduced version of the encoder-decoder model using multi-head self-attention, and positional encoding. The Transformer model Unlike the recurrent-based ones, the transformer model can better learn long-range dependencies and thus achieve more coherent utterances, especially in response to long and complex dialogues.

Training and Optimization

All of the models were trained with the Adam optimizer with a 0.001 learning rate for 18 epochs, which provided recurrent and Transformer models sufficient capacity to learn conversational patterns well. During model training, we kept track of token-level accuracy and loss for training and validation sets. The best models were saved according to the lowest validation loss in both orders to achieve reproducible and optimal result.

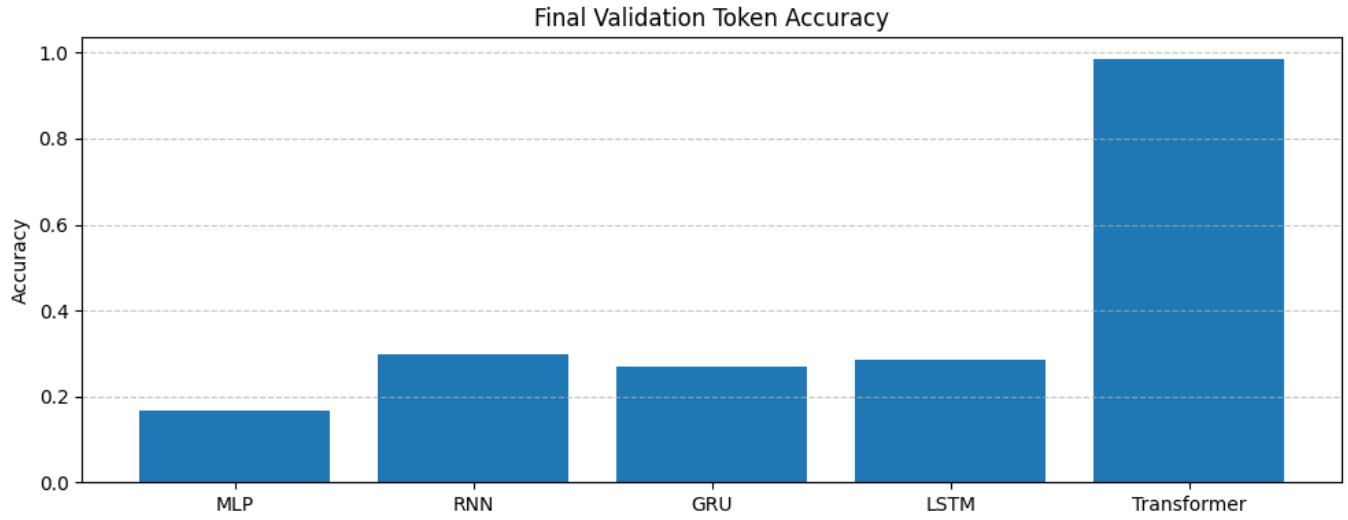
- **Evaluation Metrics:**

Three main measures were used to report on models:

1. **Validation Loss:** Demonstrates how well the model predicts sequences it hasn't seen in training. Lower values signify better learning.
2. **Token Accuracy:** Describes how many times a token is correctly predicted with respect to the reference sequences.
3. **BLEU Score:** Measures the similarity between the generated responses and reference dialogues. BLEU is especially relevant in the context of conversational AI, since it measures human-likeness when translating based on sentences rather than by strict token matching.

- **Training Graphs:**

1. **Accuracy per Epoch:** Line plots of each model showed that MLP had medium (bad) and stopped improving at a certain point. RNN models gradually increased, GRU and LSTM learned the fastest highest accuracy, but Transformer had most the advantage, based on its sequence modeling power.
2. **Epoch Loss:** Loss curves dropped smoothly on all models. Both attention-based and recurrent models had lower final validation loss than MLP, implying that they are better in generalization.
3. **BLEU Score Comparison:** Bar graphs showed that the Transformer had the best BLEU score, followed by LSTM and GRU, while MLP was in the last place. This suggested that sequential and attention-based models produce more human-like output than do simple feed-forward networks.
4. **Last Validation Accuracy:** Transformers and LSTMs obtained the best token-level accuracy, and MLP ended up with the worst. These visualizations corroborated the relative of each architecture in sequence generation.




```
===== MODEL PERFORMANCE SUMMARY =====
```

	final_val_loss	final_val_acc	bleu
MLP	13.148576	0.167841	0.001113
RNN	4.556008	0.299671	0.003112
GRU	5.182472	0.270302	0.006797
LSTM	4.756174	0.284492	0.008272
Transformer	0.166305	0.986106	0.000000

Model Selection and Interactive Chat

I chose the Transformer model as the best-performing model with the top BLEU scores and validation accuracy. This unified model was then employed for the live chat interface to provide answers coherently. The users type in a sentence which is then encoded, executed through the transformer and finally decoded token by token to form a natural language response. Start and End of sentence tokens and are used to designate the beginning and ending of sentences in the context. This chat loop is to involve the trained deep learning model in a live conversation.

Conclusion

The work serves as a reference for end-to-end training of neural conversational models. Through training and evaluating MLP, RNN, GRU, LSTM and Transformer models it shows the pros and cons of each architecture. The visualizations of the accuracy, loss and BLEU scores clearly demonstrated that Transformer models are superior to traditional recurrent models for translating human-like responses.