
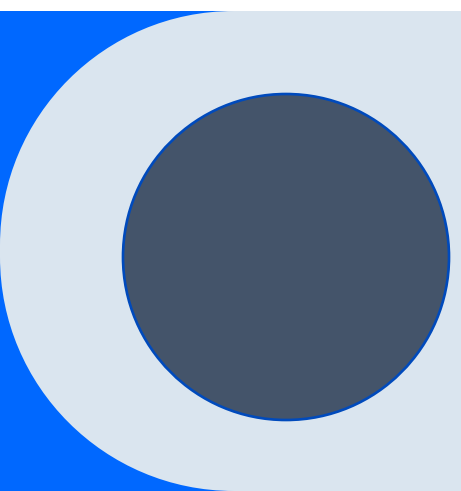


LSTM-Based Question Answering System Leveraging SQuAD 2.0

**Manushi
Ameya Santosh Gidh
Narayana Sudheer Vishal Basutkar**



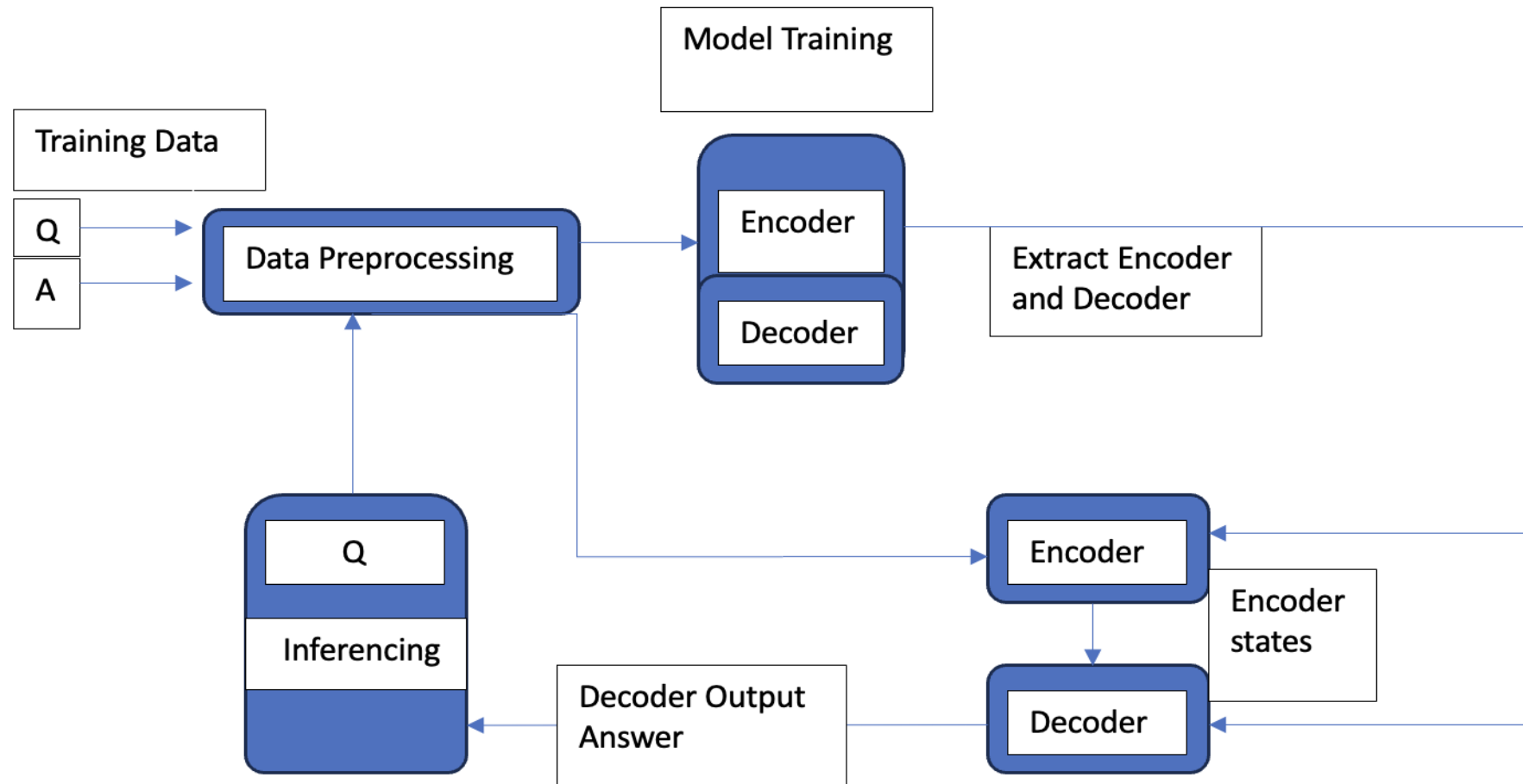
Introduction

The evolution of Artificial Intelligence (AI) has significantly advanced the capabilities of conversational systems, particularly in the realm of natural language processing (NLP). This project delves into the development of a sophisticated Question Answering Bot, utilizing the Sequence-to-Sequence (Seq2Seq) model with Long Short-Term Memory (LSTM) networks, trained on the Stanford Question Answering Dataset (SQuAD) 2.0. Our methodological approach encompasses a meticulous data preprocessing pipeline, a robust model training regimen employing k-fold cross-validation to combat overfitting, and a comprehensive ablation study to fine-tune model configurations. We aim to demonstrate how these methodological choices contribute to the enhanced performance and reliability of conversational AI applications.

Data Set

The project leverages the SQuAD 2.0 dataset, a benchmark in natural language processing, featuring 100,000+ question-answer pairs from Wikipedia. Notably, SQuAD 2.0 introduces unanswerable questions, enhancing realism. This challenges the model not just to find answers but also to recognize when none are viable, promoting robustness. SQuAD 2.0's diversity and structure make it ideal for training our Question Answering Bot, ensuring adept handling of real-world questioning scenarios.

System Architecture



Data Preprocessing

- Processes the data by tokenizing and encoding questions and answers
- Calculates the maximum sequence lengths and the vocabulary size
- Pads the sequences and prepares the data for input into the model

Model Training

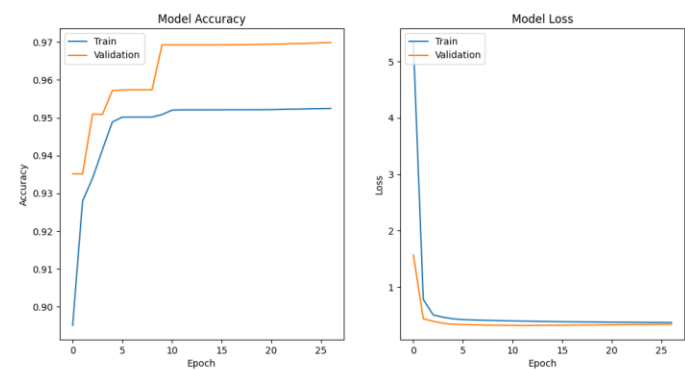
- Kfold cross validation for hyperparameter tuning and model evaluation. (K=5)
- Best hyperparameters: Dropout = 0.2, Optimizer = Adam with learning rate 10^{-5}
- Seq2Seq Model created with 1 embedding and 1 LSTM layers in each with dropouts
- Set training epochs to 100 with early stopping (stopped at 30 epochs)
- Model Accuracy = 95%

Inference and Interface

- Parameters obtained from Data preprocessing Step are used to create an Inference Object named `response_predictor`
- To predict the output sequence, we use the `model.predict` function which takes in the question text as an input and predicts the output
- We then generate a User-Friendly Interface which displays the chat history and answer for the user question

Results

Learning Curve Analysis



Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
5	0.4119	0.9511	0.4325	0.9500
10	0.3777	0.9559	0.4153	0.9541
15	0.3630	0.9559	0.4144	0.9541
20	0.3543	0.9559	0.4182	0.9542
25	0.3485	0.9563	0.4228	0.9545
30	0.3439	0.9565	0.4268	0.9546



Thank you