Implementation of a Chatbot using Jupyter Python

Submitted to: Dr. Aboozar Taherkhani

Babu Pallam (P2849288)
MSc. Artificial Intelligence

CSIP5103 – Neural Systems and NLP
November 18, 2024

Project Objective and Scope

- **Objective:** Developing a chatbot and enchancing its capabilities.
- **Scope:**
  - Implement sequence-to-sequence model using RNN with Luong attention mechanism.
  - Perform hyperparameter tuning using Bayesian optimization.
  - Experiment with alternate architectures like CNN and MLP.
  - Compare and contrast AI and non-AI approaches for chatbot functionality.
  - Explore integration with external datasets and business-related databases for practical application.

Dataset Overview and Selection

- **Primary Dataset:** Cornell Movie-Dialogs Corpus [2]
- **Additional Databases and Models:**
  - Microsoft DialoGPT Models: Leveraged as pre-trained models (small and large) from Hugging Face.
  - Persona-Chat Dataset: Used to provide diverse and personalized conversational data.
  - Custom SQL Database: Contains product information on smart home equipment.
- **Challenges:**
  - Ensuring dataset diversity.
  - Data preprocessing methods.
  - Computation resources for training of the models

Data Processing Pipeline

- **Data Cleaning and Normalization:** Convert text to lowercase, remove punctuation and special characters.
- **Stopword Removal:** Filter common words to reduce noise.
- **Lemmatization and Stemming:** Standardize words to their base/root forms.
- **Tokenization:** Split sentences into individual word tokens.
- **Data Augmentation:** Introduce diversity by randomly shuffling words in sentences.

Model Implementation Approaches (Part 1)

- **Approach 1: Base Model**
  - RNN-based sequence-to-sequence with Luong attention mechanism.
  - **Strength:** Straightforward to implement, providing a reliable baseline.
  - **Drawback:** Limited in capturing complex, long-term dependencies.
- **Approach 2: Best Tuned Model with Hyperparameter Tuning**
  - RNN-based sequence-to-sequence with Luong attention.
  - Bayesian Optimization on set of hyperparameters.
  - **Strength:** Fine-tuning improves model accuracy and response quality.
  - **Drawback:** High computational cost due to extensive tuning process.

Model Implementation Approaches (Part 2)

- **Approach 3: CNN-Based Model**
  - CNN encoder-decoder model without attention mechanism.
  - **Strength:** Faster training due to efficient parallelization in convolutional layers.
  - **Drawback:** Lacks an attention mechanism, affecting sequence coherence accuracy.
- **Approach 4: MLP-Based Model**
  - MLP encoder-decoder model without attention mechanism.
  - **Strength:** Simple architecture with fewer parameters, lowering computational needs.
  - **Drawback:** Limited ability to handle sequential data effectively.

Non-AI Methods and Alternate Database Approaches

- **Approach 5: Non-AI Methods**
  - Rule-Based Bot
    - **Strength:** Provides predictable and structured responses.
    - **Drawback:** Unable to handle complex conversations.
  - Rule-Based with Cornell Movie-Dialogs Corpus
    - **Strength:** Enhancing response.
    - **Drawback:** Still constrained by rule-based limits.
  - Menu-Based Chatbots
    - **Strength:** User-friendly navigation, quick access to option.
    - **Drawback:** Cannot address queries outside db.
- **Approach 6: Use of Alternate Database**
  - Model: RNN-based sequence-to-sequence with Luong attention mechanism.
  - Database: Persona-Chat.
  - **Strength:** Persona-Chat adds personalized responses, enhancing conversational quality and depth.
  - **Drawback:** Requires significant computational resources.

Transformer and Transfer Learning Approaches

- **Approach 7: Transformer-Based Model**
  - Model: Microsoft's DialoGPT-large, implemented using Hugging Face datasets.
  - **Strength:** Leverages a large pre-trained transformer, providing high-quality, context-aware responses.
  - **Drawback:** Computationally intensive.
- **Approach 8: Transfer Learning Approach**
  - Model: Microsoft's DialoGPT-small from Hugging Face datasets[4].
  - Integrated with SQL database of product collection related to smart home equipment.
  - UI Integration with Gradio[3].
  - **Strength:** Allows customized, data-driven responses by blending pre-trained language model with specific product information.
  - **Drawback:** May require additional fine-tuning to balance general conversational ability with specialized knowledge.

Performance

Bayesian Optimization

- **Parameters Optimized:**
  - Batch size, hidden size, learning rate, dropout, number of layers, teacher forcing ratio, gradient clipping.
- **Goal:** Improve response accuracy and coherence across models.

Evaluation Metrics

- **BLEU Score:** Measures the accuracy of generated responses by comparing with reference responses.
- **Cosine Similarity:** Evaluates the relevance of responses based on vector similarity.
- **Other Metrics can be employed:** User satisfaction through survey.

Results and Observations

- **Key Observations:**
  - Hyperparameter tuning can significantly improved RNN-based model performance.
  - Further analysis is required to assess the impact of larger datasets on model performance.
  - CNN and MLP models showed moderate response relevance without attention.
  - Computational power was identified as a key challenge due to the resource-intensive nature of model training.
  - Transfer learning with DialoGPT models led to enhanced conversational quality and coherence.
  - See Appendix for Plots and Results

For Practical Usage and Integration

- **Database Integration:**
  - Linked chatbot to SQL database for real-time product information on smart home equipment.
- **Use Cases:**
  - Customer support automation, FAQ, product recommendations.
  - Real-time response with business-related data.
  - (See Appendix for More)

Conclusion and Future Work

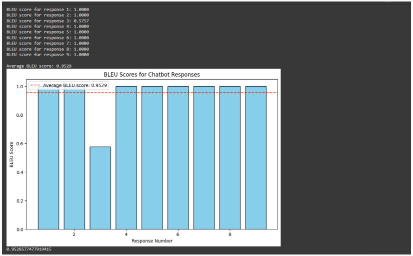
- **Conclusion:** Developed an enhanced chatbot and have done several experimentation.
- **Future Work:**
  - Implementation of more advanced architectures, like Transformers with large-scale datasets.
  - Expansion of database connectivity for specialized customer support.
  - Further fine-tuning with evolutionary algorithms.

References

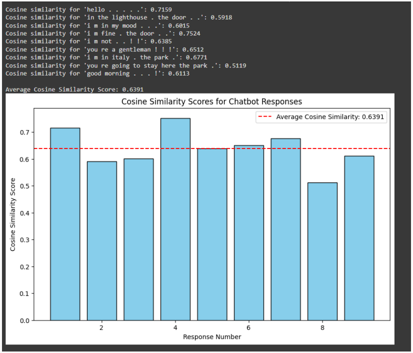
- Inka, M. (2024). *Chatbot Tutorial* [Online]. PyTorch Tutorials. Retrieved from `https://pytorch.org/tutorials/beginner/chatbot_tutorial.html`
- Danescu-Niculescu-Mizil, Cristian, and Lillian Lee. **Cornell Movie-Dialogs Corpus.** 2011. Available at: `https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html`.
- Abid, A., Farooqi, M., Zou, J. (2020). *Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild*. Retrieved from `https://gradio.app`
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J. (2020). *Transformers: State-of-the-Art Natural Language Processing*. In *Proceedings of the 2020 Conference on*

Appendix: Figures of Each Approach

Appendix: Approach 1 - Base Model

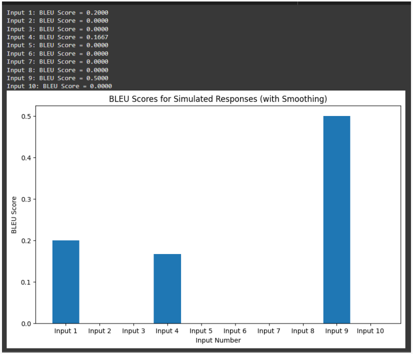


BLEU Score for Base Model

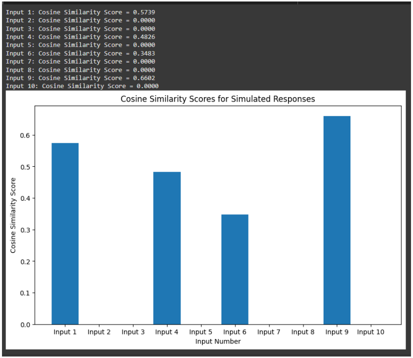


Cosine Similarity for Base Model

Appendix: Approach 2 - Hyperparameter Tuned Model

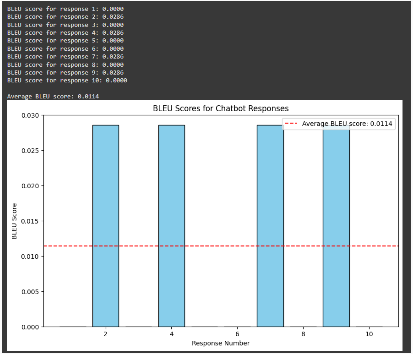


BLEU Score for Hyperparameter Tuned Model

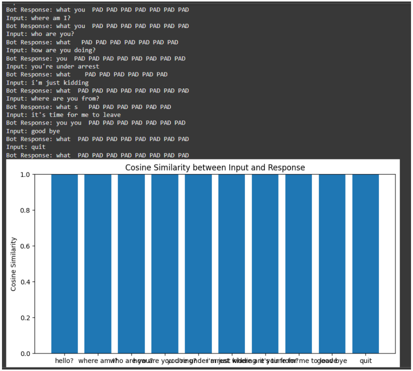


Cosine Similarity for Hyperparameter Tuned Model

Appendix: Approach 3 - CNN Model

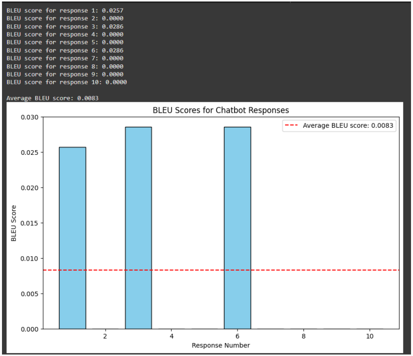


BLEU Score for CNN Model

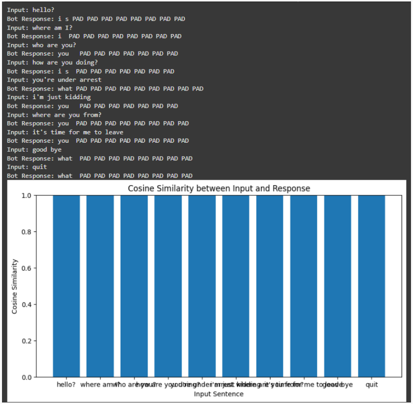


Cosine Similarity for CNN Model

Appendix: Approach 4 - MLP Model

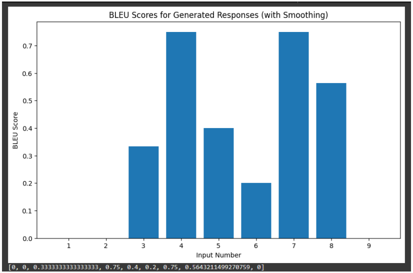


BLEU Score for MLP Model

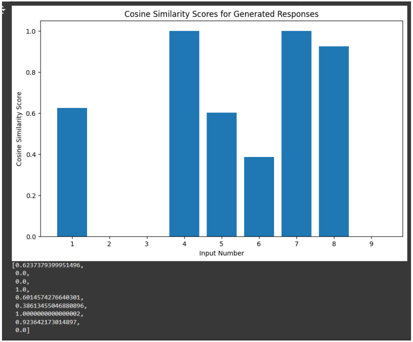


Cosine Similarity for MLP Model

Appendix: Approach 5 - Transformer-Based Model

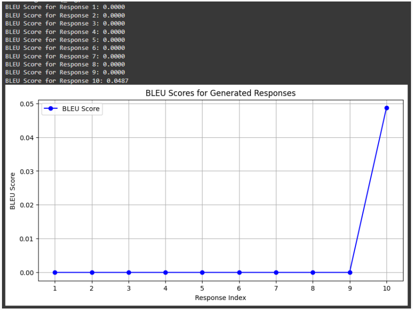


BLEU Score for Transformer-Based Model

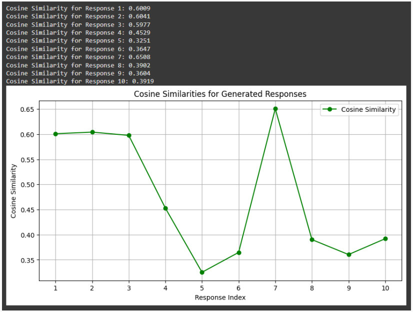


Cosine Similarity for Transformer-Based Model

Appendix: Approach 6 - Transfer Learning Model



BLEU Score for Transfer Learning Model



Cosine Similarity for Transfer Learning Model

Appendix: Individual Analysis of Each Approach

Appendix: Individual Analysis of Each Approach

1. Baseline Model (Approach 1)

BLEU Scores:

- The baseline model achieved high BLEU scores for most responses, with an average of **0.9529**.
- This indicates a strong structural and content match to the reference text.

Cosine Similarity Scores:

- Moderate scores, averaging **0.6391**, suggesting decent semantic alignment.

Summary:

- Good structural accuracy (high BLEU) but room for improvement in semantic similarity.

Appendix: Individual Analysis of Each Approach

3. CNN-Based Model (Approach 3)
BLEU Scores:

- Low average of **0.0114**, suggesting limited structural similarity.

Cosine Similarity Scores:

- Consistently **1.0000**, suggesting high semantic similarity.

Summary:

- Perfect semantic similarity, but struggles with structural alignment.

4. MLP-Based Model (Approach 4)
BLEU Scores:

- Lowest BLEU score, averaging **0.0083**, indicating minimal

Model Comparison Summary

Model	BLEU	BLEU Interpretation	Cosine	Cosine Interpretation
Baseline	0.9529	Strong structural	0.6391	Moderate semantic
Fine-Tuned	0.0867	Poor structural	0.2065	Weak semantic
CNN-Based	0.0114	Minimal structural	1.0000	Perfect semantic
MLP-Based	0.0083	Minimal structural	1.0000	Perfect semantic

END