PROJECT ABSTRACT

Chatbot using Seq2Seq

A sequence-to-sequence chatbot with Bahdanau attention trained on the Cornell Movie Dialogs corpus, delivered as a built-to-order student project with code, training notebook, report, PPT and viva kit.

01. Title

Chatbot using Seq2Seq

02. Introduction / Background

Rule-based chatbots break the moment a user phrases something unexpectedly, because every pattern must be hand-written. Sequence-to-sequence (seq2seq) encoder-decoder networks learn conversational mapping directly from dialogue data: an encoder compresses the user's message into hidden states and a decoder generates a reply one token at a time. With Bahdanau attention the decoder can focus on the relevant input words at each generation step instead of relying on a single fixed context vector. The Cornell Movie Dialogs corpus (Danescu-Niculescu-Mizil & Lee, 2011) — 220,579 conversational exchanges between 10,292 pairs of movie characters, 304,713 utterances from 617 films — provides a large, public source of prompt-response pairs for training such a model.

03. Problem Statement

Building a chatbot that responds to free-form user input without hand-written rules requires a model that maps arbitrary utterances to fluent replies. The project addresses this by training a seq2seq encoder-decoder with attention on movie-dialogue pairs, and by making the model's behaviour inspectable through attention heatmaps, beam-search hypothesis lists and perplexity-tracked training — so the student can demonstrate, evaluate and honestly discuss a learned dialogue system.

04. Objectives

- Build a prompt-response corpus of ~96,000 pairs from the Cornell Movie Dialogs corpus with reproducible cleaning, tokenization and vocabulary construction (20,000 tokens).
- Train a 2-layer bidirectional GRU encoder + 2-layer GRU decoder with Bahdanau attention using teacher forcing, tracking train/validation loss and perplexity per epoch.

- Implement greedy and beam-search (width 5) decoding with an attention heatmap visualizer for any prompt-reply pair.
- Deliver the complete student kit: source code, training notebook, trained weights, report, PPT and viva Q&A.

05. Proposed Methodology

The corpus is parsed into consecutive prompt-response pairs, filtered to a maximum of 20 tokens per side and mapped through a 20,000-word vocabulary with [start]/[end] tokens. The bidirectional GRU encoder produces per-token hidden states; the decoder is trained with teacher forcing (ratio annealed from 1.0 to 0.5) to predict each reply token from the previous ground-truth token and the attention-weighted encoder context, minimizing cross-entropy with the Adam optimizer (learning rate 1e-3, gradient clipping at 5.0). Validation perplexity drives early stopping, and the best checkpoint is kept for the chat demo.

06. System Architecture / Working

The system has three stages. (1) Data pipeline: corpus parsing, cleaning, tokenization, vocabulary building and padded batching. (2) Model: embedding layer (300 dimensions), 2-layer bidirectional GRU encoder (512 hidden units), Bahdanau additive attention, and 2-layer GRU decoder. (3) Application: a chat UI feeds the user message through the encoder and generates the reply autoregressively until [end] or a length cap; a separate analysis view renders attention heatmaps and ranked beam hypotheses with log-probabilities.

07. Hardware / Software Requirements

- Python 3.10, PyTorch, NLTK, NumPy, Matplotlib, Jupyter, Flask
- CPU training is supported; a CUDA GPU shortens training from many hours to a few
- Cornell Movie Dialogs corpus (public download, one-time internet access)
- 8 GB RAM recommended

08. Expected Outcomes

A working chatbot that responds to free-form input with fluent, movie-dialogue-styled replies; a training notebook that logs loss and perplexity curves from the student's own run; attention visualizations showing word-level alignment; and an honest error analysis documenting the model's known tendency toward safe, generic replies. No fixed quality metric is claimed — the reproducible training experiment is the outcome.

09. Applications

- Teaching aid for NLP coursework: tokenization, RNNs, attention, decoding strategies

- Baseline for comparing dialogue approaches (retrieval vs generative, GRU vs Transformer)
- Demo frontend for conversational-AI project exhibitions
- Starting point for domain-specific retraining (e.g. college FAQ corpus)

10. Future Scope

- Multi-turn conversation memory across dialogue turns
- Transformer-based encoder-decoder for longer context
- Diverse decoding (top-k / nucleus sampling) to reduce generic replies
- Retrieval-augmented responses grounded in a knowledge base