

Airline Chatbot Project Report -Richa Patel

Overview: This project developed an interactive airline chatbot using FLAN-T5, fine-tuned on `chatbot_data.tsv` (5,521 pairs after filtering) for a major airline's customer service. Data was sourced from Kaggle datasets and FAQ pages of multiple airlines (e.g., booking, baggage, pet policies), converted from `.txt` to `.tsv`. The chatbot features fuzzy matching for context-aware responses, BLEU score evaluation (0.362), and a real-time interactive loop, built in Google Colab's T4 GPU environment. Training was visualized via Weights & Biases (W&B) at <https://wandb.ai/richricha4939-univerai/huggingface/runs/kj0i5oab> (validation loss 0.0015).

Data and Preprocessing: The dataset (`chatbot_data.tsv`, originally ~11,576 pairs) was filtered to 5,521 airline-specific pairs (4,416 training, 1,105 validation) using regex (e.g., `flight|airline|baggage|ticket|check-in|pet|seat`). Preprocessing included NaN removal, deduplication, and T5 tokenization. An earlier run used 1,576 pairs, limiting response diversity.

Architecture and Training: The chatbot uses FLAN-T5-base, a Seq2Seq model, fine-tuned with Hugging Face Trainer. Fuzzy matching (`fuzz.token_sort_ratio`) enhances context by retrieving similar dataset entries. Training achieved a validation loss of 0.0015, logged via W&B, though earlier runs used 1,576 pairs.

W&B Training Loss Plot (Validation Loss: 0.0015)

```
FutureWarning: The `use_auth_token` argument is deprecated in favor of `token` argument. Please use `token=use_auth_token` as a one-off argument for this warning. wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server) wandb: You can find your API key in your browser here: https://wandb.ai/authorize?ref=models wandb: Paste an API key from your profile and hit enter: wandb: Paste an API key from your profile and hit enter: wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or running `wandb login` from the command line wandb: No netrc file found, creating one. wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc wandb: Currently logged in as: richricha4939 (richricha4939-univerai) to https://api.wandb.ai. Use `wandb login --relogin` to force a new login. Tracking run with wandb version 0.19.10 Run data is saved locally in /content/wandb/run-20250506_182723-kj0i5oab Syncing run ./flan-t5-finetuned to Weights & Biases (docs) View project at https://wandb.ai/richricha4939-univerai/huggingface View run at https://wandb.ai/richricha4939-univerai/huggingface/runs/kj0i5oab Passing a tuple of `past_key_values` is deprecated and will be removed in Transformers v4.48.0. You should pass an instance of `EncoderDecoderModel` instead. [2844/3312 17:51 < 02:56, 2.65 it/s, Epoch 2.58/3]
```

Epoch	Training Loss	Validation Loss
1	0.010300	0.002298
2	0.004000	0.001789

[3312/3312 21:21, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	0.010300	0.002298
2	0.004000	0.001789
3	0.003600	0.001513

There were missing keys in the checkpoint model loaded: ['encoder.embed_tokens.weight', 'decoder.embed_tokens.weight']

Model fine-tuned and saved.

Connected to Python 3 Google Compute Engine backend (GPU)

Results: Evaluated with BLEU score (0.362, based on 1,576 pairs), the chatbot excels for flight changes (“you can change your flight date by logging into your account...”), baggage (“the baggage allowance for international flights is 2 checked bags...”), and seat upgrades (“yes, you can upgrade your seat...”). Generic responses for pet (“[generic response]”), check-in, and some baggage queries arose due to the smaller dataset’s limited coverage. The interactive loop demonstrated real-time usability, with screenshots captured. The larger dataset (5,521 pairs) may improve performance if re-run.

Challenges: Key challenges included fixing an `eval_strategy` error for validation, resolving W&B/Hugging Face API prompts, correcting GPU device mismatch (`cuda:0` vs. `cpu`), and addressing `SettingWithCopyWarning` in fuzzy matching. The initial 1,576-pair dataset reduced fuzzy matching effectiveness, causing generic responses.

Conclusion: The chatbot effectively handles flight, baggage, and seat queries, with robust interactivity. Future work involves re-running with the 5,521-pair dataset to enhance response quality and coverage.