

# ATIS Chatbot

Laconia Team Project

---

Muqtadir A, Nikhil Madduru, Karthik Mikkelineni

May 2025


# Table of Contents

- Project Overview
- Key Features
- Technology Stack
- Dataset Preparation
- Model Development and LangChain Implementation
- Evaluation Metrics
- Training Plots
- Challenges Faced
- Next Steps
- Application Images
- Conclusion

# Project Overview

---

# Project Overview

- **Objective:** Develop an intelligent chatbot for airline travel queries using the ATIS dataset.
- **Achievements:** LangChain-based conversational flow implemented, Flask API and UI added.
- **Repo Link** 

## Key Features

---

# Key Features

- Intent classification (8 intents: abbreviation, aircraft, airfare, airline, flight, flight\_time, ground\_service, quantity).
- Entity extraction (cities, dates, airlines).
- Context-aware responses using LangChain.
- Modern Flask-based UI and server hosting, with aligned chat bubbles and "Clear Chat" functionality.

# Technology Stack

---

# Technology Stack

- Python 3.12.
- DistilBERT (intent classification).
- Flan-T5-Base and Flan-T5-XL with LoRA.
- LangChain with `langchain-huggingface` (conversational flow, LLM integration).
- Flask (server hosting and UI).
- Supporting Libraries: `transformers`, `peft`, `spacy`, `nlTK`, `sklearn`, `pandas`, `flask-session`, `sentencepiece`.



# Dataset Preparation

---

# Dataset Preparation

- **Source:** ATIS dataset (`atis_train.csv`, `atis_test.csv`).
- **Intents:** 8 intents (e.g., flight, airfare, ground\_service).
- **Entities:** Cities, dates, airlines.
- **Processing:** Handled missing data, subsampled for efficiency.

# Model Development and LangChain Implementation

---

# Model Development and LangChain Implementation

- **DistilBERT Intent Classifier:** Achieved 99.25% accuracy.
- **Flan-T5 Models:** Fine-tuned with LoRA for response generation (Flan-T5-Base and Flan-T5-XL).
- **LangChain Framework:** Conversational flow and LLM integration using `HuggingFacePipeline`.
- **Context Retention:** Maintains conversation context across multiple turns.

# Evaluation Metrics

---

## Evaluation Metrics: Base Classifier (Logistic Regression)

The base classifier (Logistic Regression) serves as a baseline.

Metric	Value
Accuracy	96.49%
Precision (weighted)	96.14%
Recall (weighted)	96.49%
F1-score (weighted)	96.16%

## Evaluation Metrics: Tuned DistilBERT Classifier

The tuned DistilBERT model was fine-tuned for intent classification.

Metric	Value
Accuracy	99.25%
Precision (weighted)	99.55%
Recall (weighted)	99.25%
F1-score (weighted)	99.35%

# Comparison: Base vs. Fine-Tuned DistilBERT Classifier

Comparison of core metrics for intent classification:

Metric	Logistic Regression	DistilBERT	Difference
Accuracy	96.49%	99.25%	+2.76%
F1-score (weighted)	96.16%	99.35%	+3.19%



## Training Plots

---

# DistilBERT Loss Plots

Loss plots provide insight into the DistilBERT intent classifier training process.

- Training Loss Plot:

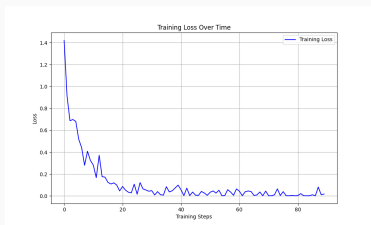


Figure 1: Training Loss plot

- Eval Loss Plot:

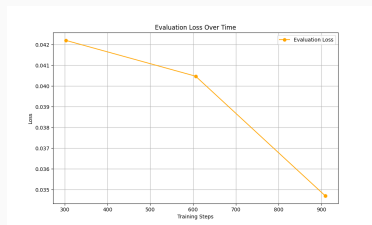


Figure 2: Eval Loss plot

# LoRA Training Plots

LoRA (Low-Rank Adaptation) was used to fine-tune Flan-T5 models for response generation.

- Flan-T5-Base:

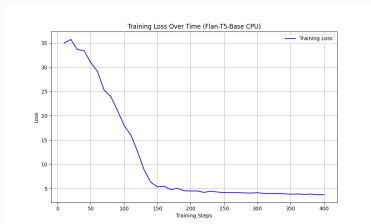


Figure 3: CPU Loss plot

- Flan-T5-XL:

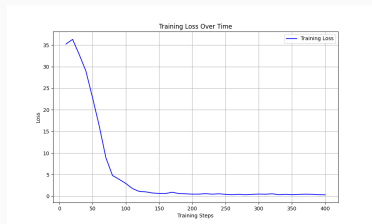


Figure 4: GPU Loss plot

## Challenges Faced

---

## Challenges Faced: Functional Issues

- **City Order Swapping:** Misinterpreted departure and destination cities in queries.
- **Context Retention Issues:** Difficulty retaining specific entities after invalid inputs.
- **Intent Misclassification:** Occasional confusion between similar intents.

# Challenges Faced: Rasa Incompatibility

- **Initial Approach:** Project initially used Rasa 3.6.21 for conversational flows.
- **Incompatibility Issues:**
  - Dependency conflicts with **torch**, **numpy**, and other libraries.
  - Challenges integrating Rasa with python and libraries and with intent classifier.
- **Solution:** Switched to LangChain for better compatibility and flexibility with LLMs.

## Next Steps

---

## Next Steps

- Test intent and entity accuracy.
- Enhance Flask UI (e.g., user feedback).
- Develop a robust summarizer (e.g., for flight details).
- Improve NER mappings (e.g., for better entity extraction).



## Application Images

---

# Application Screenshots

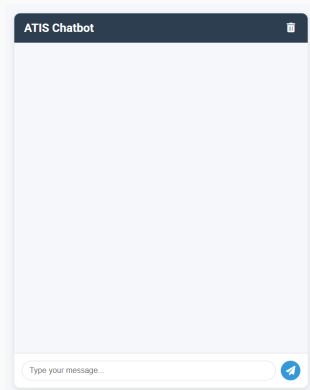


Figure 5: Chatbot Interface

# Working Test Samples for Multi-Turn Conversation

- **Sample 1: Flight Booking**

- **User:** I want to fly from Boston to Denver tomorrow.
- **User:** Morning flight, before 9 AM.
- **User:** Business class available?
- **User:** Any nonstop flights?

- **Sample 2: Fare Inquiry with Ground Service**

- **User:** Cheapest airfare from Boston to Denver next week.
- **User:** Any deals for economy class?
- **User:** Ground transportation options in Denver airport.

# Application Screenshots (Cont.)

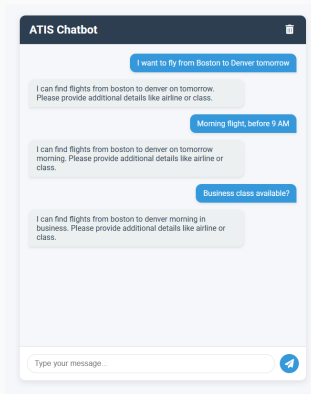


Figure 6: Multi-Turn Interaction

# Conclusion

---

# Conclusion

- **Summary:** Built a robust chatbot with LangChain, achieving 99.25% intent classification accuracy.
- **Future:** Continue testing, source more dialogues for fine tuning, enhance UI features.