Hackathon Task: Few-Shot/Zero-Shot Car Sales Conversation Information Extractor and Analyzer

Background

Car sales conversations contain crucial information about customer preferences and concerns. This task aims to develop a system that can automatically extract key information from car sales conversation transcripts using few-shot or zero-shot learning approaches, making it easier to analyze customer requirements and improve the sales process.

Objective

Create a tool that can process car sales conversation transcripts and extract specific types of information related to customer requirements, company policies discussed, and customer objections without extensive training data.

Constraints

1. No use of external APIs for Large Language Models (LLMs) is allowed.
2. Teams can have a maximum of 3 members. Individual participants are also welcome.

Pre-requisites

- Participants should preferably have access to a CUDA-enabled machine or a Colab/Kaggle account to use GPU-powered notebooks for development.
- Necessary libraries should be pre-installed on their machines (e.g., PyTorch, TensorFlow, NLTK, spaCy).

Requirements

1. Input: The system should accept conversation transcripts in plain text or PDF format.
2. Processing: Implement natural language processing techniques to analyze the text and extract relevant information.
3. Output: Generate a structured output (JSON) containing the extracted information.

4. Frontend: Develop a simple web interface where users can upload transcript files and view results.
5. Participants will be provided with a set of transcriptions to develop the pipeline, and their solution will be evaluated on similar set of hidden test files.

Information to Extract

1. Customer Requirements for a Car:
    - Car Type (Hatchback, SUV, Sedan)
    - Fuel Type
    - Color
    - Distance Travelled
    - Make Year
    - Transmission Type
2. Company Policies Discussed:
    - Free RC Transfer
    - 5-Day Money Back Guarantee
    - Free RSA for One Year
    - Return Policy
3. Customer Objections:
    - Refurbishment Quality
    - Car Issues
    - Price Issues
    - Customer Experience Issues (e.g., long wait time, salesperson behaviour)

Output Format

The extracted information should be in a structured JSON format with separate keys for each detail. If a detail is not found in the conversation, its value should be null.

Submission

Submissions should be made through a GitHub repository containing:

1. All code and documentation
2. A submission JSON file containing results for all test transcripts, with each transcript identified by its assigned conversation-ID.

Evaluation Criteria

1. Accuracy of Customer Requirements extraction - 20 points
2. Accuracy of Company Policy extraction - 10 points

3. Accuracy of Customer Objections extraction - 20 points
4. Creativity in approach and problem-solving - 30 points
5. Code quality, organization, and documentation - 20 points

Resources

- Natural Language Processing libraries (e.g., NLTK, spaCy)
- PDF parsing libraries (e.g., PyPDF2, pdfminer)
- Web framework for frontend (e.g., Flask, Django for Python; or React for a separate frontend)
- Data visualization libraries (e.g., Matplotlib, Plotly)
- Sample car sales conversation transcripts for testing (only test set provided, no training data)

Bonus Tasks

1. Bulk Upload and Analysis:
    - Implement functionality to handle bulk upload of multiple conversation transcripts in a single file, separated by a delimiter.
    - Provide an analysis dashboard with visualizations (e.g., bar charts) showing:
        - Distribution of most requested car colors
        - Popular price ranges
        - Preferred car types
        - Common refurbishment issues
        - Frequently raised objections
2. Export Functionality:
    - Allow users to export analysis results and visualizations in common formats (e.g., PDF, CSV)

Prizes

Winning team will take home a INR 25k Cash Prize 31st August and a chance of Internship/PPO.

* INR 5k additional Cash Prize for completing the **bonus task**, prerequisite is to complete the 1st task.