INTENT DETECTION

Goal 1: Framing the Problem as a Machine Learning Task

1. About the Dataset

- The dataset consists of text queries with corresponding intent labels (e.g., EMI, Warranty, Delivery). The task is to classify these queries into predefined categories, making it a multi-class classification problem.

2. Data Preprocessing

- Text Cleaning: Remove special characters, extra spaces, and convert text to lowercase.
- Tokenization: Break down sentences into words/subwords using BERT's pre-trained tokenizer to convert text into numerical format.
- Padding and Truncation: Ensure all text sequences are of the same length by padding or truncating the text.
- Encoding Labels: Convert intent labels (e.g., EMI) into numeric values using LabelEncoder.
- Data Splitting: Split the data into 80% training and 20% testing sets for model training and evaluation.

3. Train and Test Data

- Training Data: Used for model learning, containing both input queries and corresponding labels.
- Test Data: Used to evaluate the model's ability to generalize to unseen queries.

4. Model Selection: Why BERT?

- Contextual Understanding: BERT captures the full context of each word in a sentence, improving understanding of varied user queries, which traditional models like Naive Bayes cannot handle.
- Pre-trained Model: BERT is pre-trained on a large corpus and can be fine-tuned for intent detection without starting from scratch.
- State-of-the-Art Performance: BERT outperforms traditional models in tasks like text classification and intent detection, especially when considering sentence context.

5. Why Not Traditional Models?:

- Traditional models, such as Logistic Regression, Naive Bayes, or SVM, fail to capture the complexity relationship in text, particularly while considering the context. Even though these models are faster and computationally cheaper, they haven't performed as well as BERT in tasks such as intent detection, where full sentence context plays a vital role.

6. Model Evaluation

- Metrics:
  - Accuracy: Percentage of correct predictions.
  - Precision, Recall, F1-Score: Evaluate model performance for each intent, balancing between false positives and false negatives.
- Evaluation Approach: We evaluate the model on the test set to assess its ability to generalize to unseen data.

Goal 3 : Justification of Results and Model Improvements

- The results obtained during the evaluation phase (e.g., accuracy of 50%) indicate that the model is performing reasonably well for some intents but may be struggling with others. While 50% accuracy is not ideal, it may reflect several aspects of the dataset and model training process.

Why the Results Make Sense:

- **Data Complexity**: Intent detection often involves capturing subtle variations in phrasing. Even minor differences in sentence structure can result in different predictions. The dataset may have queries that are semantically similar but belong to different classes.
- **Model's Generalization**: Since the model is trained on a limited dataset, it might be unable to generalize effectively on unseen data, especially if certain intents are underrepresented in the training set.

Future Improvements:

1. **Data Augmentation**: Expanding the dataset through paraphrasing or augmenting with additional queries will help the model generalize better.

2. **Hyperparameter Tuning**: Adjusting the learning rate, batch size, and number of epochs will improve the training process and model accuracy.

3. **Class Weighting**: To handle potential class imbalance, adjusting the class weights will ensure fairer classification across all intents.