1. Describe how you implement your model, including your choice of packages, model architectures, model input, loss functions, hyperparameters, etc.

   For the packages I use, I used hugging face transformers for model loading, tokenization and training. And PyTorch for underlying framework for deep learning operations, scikit-learn for metrics like precision, recall, F1-score, and accuracy. And finally NLTK for data augmentation techniques like synonym replacement.
   For the model architecture, I used RoBERTa-large, which is a transformer based model with a strong ability to handle language understanding tasks. Also I used output layer to predict the response quality.
   For the model input, I combine the utterance, situation, and response.
   For the loss function, I used cross-entropy loss, which is handled internally by the Trainer for classification tasks.
   Finally, for the hyperparameters, learning rate is 2x10^5, batch sizes is 8 for training, 16 for evaluation, warmup steps is 500, weight decay is 0.01. Two epochs are run to get the model.

2. What processing did you do with the data? Is there an improvement in predictive accuracy when utilizing both situations and utterances for prediction, compared to solely relying on utterances? Why or why not?

   I combined the utterance, situations, and responses into a structured input format, and I do data augmentation like synonym replacement and random deletion to enhance generalization.
   There is indeed an improvement than solely relying on utterances, this is because that utterances alone might not provide sufficient context for accurately predicting response quality. However, with situations combined, the model can have better comprehension of the current situation and have a better acknowledgement of what the utterances actually mean, thus can have better results.

3. Compare all the methods you have tried and use a table to display their respective performances. Which method performed the best, and why?

Method	Performance (percentage of correct)
BERT with no augmentation	0.72
Give GPT-4-turbo the utterance and response and ask it to give me the prediction	0.68
BERT with augmentation	0.74
RoBERT-a large with augmentation	0.805

We can see that RoBERT-a with data augmentation has the best result, because RoBERT-a large is a stronger model compared with simple BERT, and data augmentation actually helps increasing the comprehension for the model about the language.