The solution use transformer model for intent classification of ATIS dataset provided

Data preparation

- The data is multilabel separated by “+”. It was converted to a list of labels for each sample
- An Enum of intents (all intents in training data) is used to make sure the api/classifier supports a closed set of intents.
- Converted to numpy arrays before feeding to the NN.

Train

- The training fine-tunes a pre-trained model from hugging face and its trainer to train. “"distilbert-base-multilingual-cased" which is language agnostic.
- It only uses the training data and do 3-fold cross validation(ideally should do 10-fold) and then train model on all train data and save the model
- To run training script, inside script
    - PYTHONPATH=../ python train_intents.py
- The training uses gpu if available. I trained it in a AWS GPU instance and it took 2 hours for cross validation and building the final model.

Eval

- To get evaluation on test data, inside script run
    - PYTHONPATH=../ python test_eval.py
- It will print a classification report with an f-score for each intent.
- You can find the results already in results/test_eval_report.txt

API

- It uses Flask to serve the model and follows the spec specified in the challenge.
- It handles the errors properly and send response accordingly.
- To run the server, run
    - python server.py --model model/intent_model.bin

Notes

- I wrote a small test for the classifier to showcase how the code should be tested. Ideally all methods in bert_intent_classifier should be tested. And API responses for different cases should be tested as well. I didn’t do it for lack of time.
- Usually we use Dockerfile and Makefile to run everything in a container with simple commands. I kept it simple on this one because of time constraints.
- For experiments, I have seen that pre-trained transformer models for text classification works really well with a low number of samples. I usually try simpler models with tf-idf features as it’s faster to train but since this was more a NN challenge I didn’t write code for those experiments.