Name : Shraman Pal

Github repo link : <https://github.com/ShramanPal/cartography>

Google drive link: <https://drive.google.com/drive/folders/16ppOMcYwCYDCCWdTHjezNRb258RsxRmM?usp=sharing>

(There were some errors while trying to use git, so I had to add manually. In case any trouble arises while running repo, all the files are present in the gdrive.)

**PROCEDURE:**

Implemented K fold validation method using Sklearn package.

Depending on the dataset, I created a list of indices of the same length as that of the dataset. Passed it through Sklearn’s K fold implementation to generate the indices that will belong to the train set and the dev set. These indices are stored in a list for each iteration. For example if there are 4 examples [1, 2, 3, 4] and no of folds=2. Then the train indices list stores [[1, 3], [2, 4]] if [1, 3] are in the train set first and [2, 4] in the train set next. A similar list of indices for the dev set.

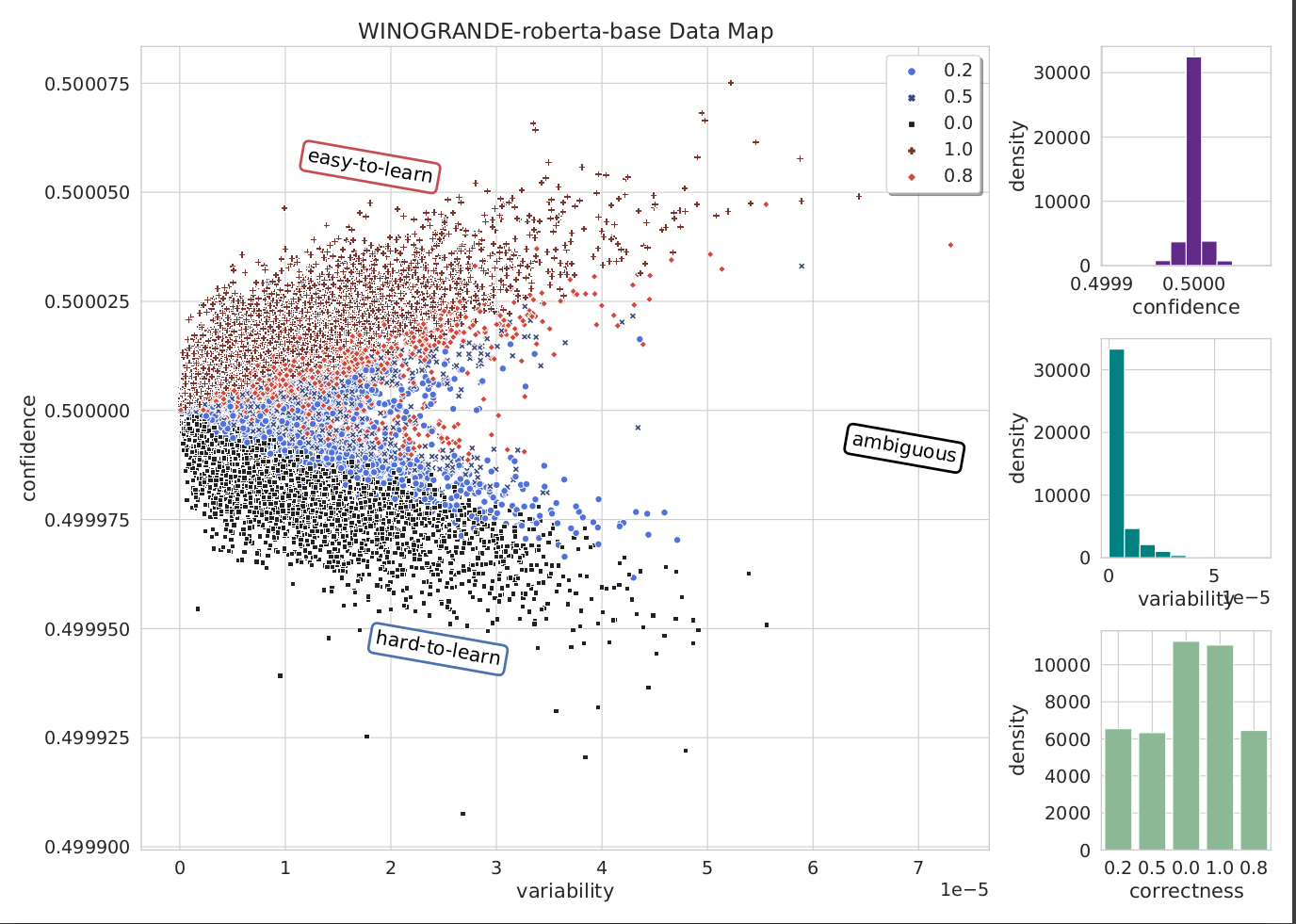
These indices are then passed in the train function and used in train\_dataloader generation (sampler=indices) to generate the train samples as the dataset is generated only once in the source code.

After training on these samples for each epoch, the model evaluates on the dev set while saving the model parameters. During this evaluation the necessary info (guid, logits, gold) of the dev set for that epoch are stored in a jsonl file similar to the train set in the original code. In this way for a particular dev set, data is stored for all epochs.

Once a new fold is chosen as the dev set, epoch counts restarts allowing all the data samples to have the same epoch range i.e all samples are stored in the same epoch file. (This allows a possible place for scrutiny as the model inevitably learns in the later folds even if the epoch count restarts. Therefore they are not judged equally during calculation).

Plots are generated using the same code with just an added argument of split (train or eval) to plot the eval dynamics or training dynamics.

**PLOT**

****

The model (roberta base) was trained for 12 epochs total (4 epochs per fold(nfolds=3)). The optimal learning rate provided was given for roberta-large model and the same was used (probably not the most optimal learning rate). The plot of confidence shows that the model has not learnt from the data as it forms a normal like distribution around the mean. The variability is packed at 0, which probably comes from the fact that it has not learnt and therefore did not change its predictions for some samples right from the start.

**THIRD TASk**

As the model could not be trained for a sufficient period of iterations as well as optimal hyperparameters were not used (roberta base) several of these examples are possibly incorrect even though it could be correct if trained upto a proper extent. Some examples which possibly fall under hard to learn categories (wrong or tough to answer):

Alice decided not to grow a vegetable garden in her backyard because the \_ is too large.

Options garden backyard

Gold: 0 Pred: 1

(the backyard could be too large to be able to maintain a garden)

Victoria gets a bottle of formula for Patricia every couple of hours because \_ is her parent.

Options Victoria Patricia

Gold: 0 Pred: 1

(Victoria could be the daughter as well and bring her parent a bottle of formula)

Kevin soon realized that Randy was autistic because \_ is very perceptive about such things.

Options Kevin Randy

Gold: 0 Pred: 1

(Model requires to understand about autism).

Brett bought the airplane tickets for Joel and their family after \_ checked their schedule.

Options Brett Joel

Gold: 1 Pred: 0

(Brett (who could be Joel’s brother) himself could have checked their schedule.