Progress Report

Three of us had a kick off meeting a few weeks ago, and decided to work separately on data cleaning and preprocessing. This decision was made because we won't be able to work on ML/DL implementation without preprocessing. We share our preprocessing code with each other, and make progressive changes on our individual works. Since there are many directions we can go with feature engineering such as whether to use just the response or use both response and context, keep the emojis or leave it alone, etc., we have kept all different versions of preprocessing and feature engineering codes to implement ML/DL models later on.

Current progress

We have put together a few different preprocessing codes, and have created ML models (like Multinomial Naive Bayes), LSTM, BiLSTM and BERT models. We have completed a whole cycle of the project such as importing the train, test file, preprocess and data cleaning, implementing DL models, and have created an "answer.txt" file as per the project requirement. We have individually submitted the code (pushed) through livelab, but failed to beat the baseline.

Remaining Task

Since we completed the whole cycle of project submission without beating the baseline, our task is to improve our models. We need to improve on feature engineering by trying a few different methods. Some of them we are thinking that could help are:

1. Leaving few punctuation marks such as "!" which could improve the model,
2. Run models without removing emojis which might help.
3. Need to figure out a perfect way to combine response and context. Such as do we use all the context or just the last one followed by the response,
4. Try pre-trained models like K-train
5. Fine tune the models (using different epoch, validation split allocation, etc)

Challenges

I. Although we achieved 0.694 of f1, we still need to improve the BERT model to beat the baseline (which is 0.723 of f1) or learn about K-train as well.
II. To find a perfect way to use response and context is one of the challenges.
III. Find the perfect feature engineering such as whether we need to remove all punctuation, emoji, stopwords, etc.