Notes

Modeling

- input is vector consisting of concatenating index vectors for:
    - reply up to current point with batch sequence length l_r
    - conversation history of up to h sentences with batch sequence length l_h
        - l_i = l_h = 10? select based on sentence length statistics
        - dimensionality is h * l_h
    - personality vector of up to p sentences with batch sequence length l_p
        - created based on genres of movies characters appear in
            - pick p based on average number of different genres movies consist of - likely 2-5
                - add small probability (e.g. .2) to randomly switch label (at test?) so chatbot is more robust to different personality combinations
        - l_p = 5? up to us
        - dimensionality is p * l_p
    - along with segment vector, padding, special tokens for which speaker, etc
- output is decoded response sentence
    - top-k decoding with k = 40
- target is correct response sentence
    - group stratified cross-validation during train test split to prevent target leakage, where group is conversation index
        - if only one row for conversation, can go in either set. if multiple, put unique index in either train or test.
- use feature representations of pretrained transformer model (gpt-2) then retrain fc layer of decoder
    - need new vocabulary for output layer?
        - Preexisting vocab is likely fine
    - will concatenated input format allow this?
        - looks like it
    - don’t retrain classifier layer (?)
- cross entropy loss with adamW optimizer

Evaluation

- use test perplexity as metric for model comparison/selection
- for ui or self-guided evaluation, users pick personality from provided set of sentences based on genre