b.py

`def create_uniform_distribution(self, name):`

The above function in `b.py` should be defined according to the following formula,

$$\forall_{e \in \boldsymbol{E}} \; \forall_{f \in \boldsymbol{E}} \;\; t(e|f) = \frac{1.0}{|\boldsymbol{F}|} \tag{1}$$

where $t(\ldots)$ is a conditional distribution with the name `name`, $\boldsymbol{E}$ is the English vocabulary, $\boldsymbol{F}$ is the foreign vocabulary, $|\boldsymbol{F}|$ is the number of words in the foreign vocabulary.

The function should return the initialized conditional distribution $t(\ldots)$.

`def conditional_probability(self, sentence_index, epsilon, conditional)`

The above function in `b.py` should be defined according to the following formula,

$$p(\boldsymbol{e}|\boldsymbol{f}) = \frac{\epsilon}{(l_f)^{l_e}} \sum_{j=0}^{(l_e-1)} \sum_{i=0}^{(l_f-1)} t(e_j|f_i) \tag{2}$$

where the function argument `conditional` is represented in the formula by $t(\ldots)$, and where $\boldsymbol{e}$ and $\boldsymbol{f}$ are the English and foreign sentences at `sentence_index`, e_j is the j^{th} token in the English sentence, f_i is the i^{th} token in the foreign sentence, l_e is the length of the English sentence in tokens, l_f is the length of the foreign sentence in tokens, and ϵ is `epsison`.

The function should return the floating point value $p(\boldsymbol{e}|\boldsymbol{f})$.

`def perplexity(self, epsilon, conditional):`

The above function in `b.py` should be defined according to the following formula,

$$PP = -\sum_{s=0}^{S-1} \log_2 p(\boldsymbol{e}_s|\boldsymbol{f}_s) \tag{3}$$

where S is the number of sentence pairs in the parallel corpus, s is a sentence index, and $p(\boldsymbol{e}_s|\boldsymbol{f}_s)$ in the formula represents a function call to `conditional_probability`.

The function should return the floating point value PP.
