Name: ZHONG Qiao Yang

NetID: 24112456g

method 1 (classification)

0. **delimiter:**

   The punctuation like `? , . \n \t` we set them as the delimiter to split sentences in rules at first.

1. **set labels:**

   we set 4 labels: **S** (start of word), **M** (mid of word), **E** (end of word) , **O** (single character is a word)

   for example, we tag the sentences `人民英雄永垂不朽`, then we can tag it with labels `SESESMME`, which means we can split it into 3 words `人民，英雄，永垂不朽`.

2. **models:**

   First, we tokenize the sentence character by character, and we feed the token into the model, and we output the logits of labels. For example,

   > 人 -> [0.4, 0.2, 0.3, 0.1]
   >
   > 民 -> [0.1, 0.2, 0.4, 0.3]

   the output stands for the prob of S M E O labels, and we can choose argmax one.

   We can apply models like **CRF, bidirectional RNN, Transformers**, and we add a classification layer to the output layer for each character embedding so that we can get RNN and Transformers for classification tasks.

method 2 (Byte Pair Encoding)

0. **delimiter:**

The punctuation like `? , . \n \t` we set them as the delimiter to split sentences in rules at first.

1. First, we split the sentence into characters. For example, `人民英雄永垂不朽` -> `人 民 英 雄 永 垂 不 朽`, and add them into `vocabulary`
2. And we compute the frequency of each pair of characters in the corpus

   > 人民 frequency = ??
   >
   > 民英 frequency = ??
   >
   > 英雄 frequency = ??
   >
   > ...

3. Select the pair with largest frequency, like `英雄`, and add it to `vocabulary`
4. We loop the step 3 until we achieve the upper bound of `max len of vocabulary` or other hyper-parameters.
5. we apply the `vocabulary` to tokenize the new sentence, we choose the word in `vocabulary` and we choose the longest one greedily. For example,

   > 人民英雄永垂不朽 人民 is the longest word in vocabulary, we split it
   >
   > and then we repeat the same manipulation from 英雄永垂不朽
   >
   > until all sentence has been seperated.