Artificial Intelligence Lab Work (5)

レポート解答用紙（Report Answer Sheet）

学生証番号（Student ID): 20521150

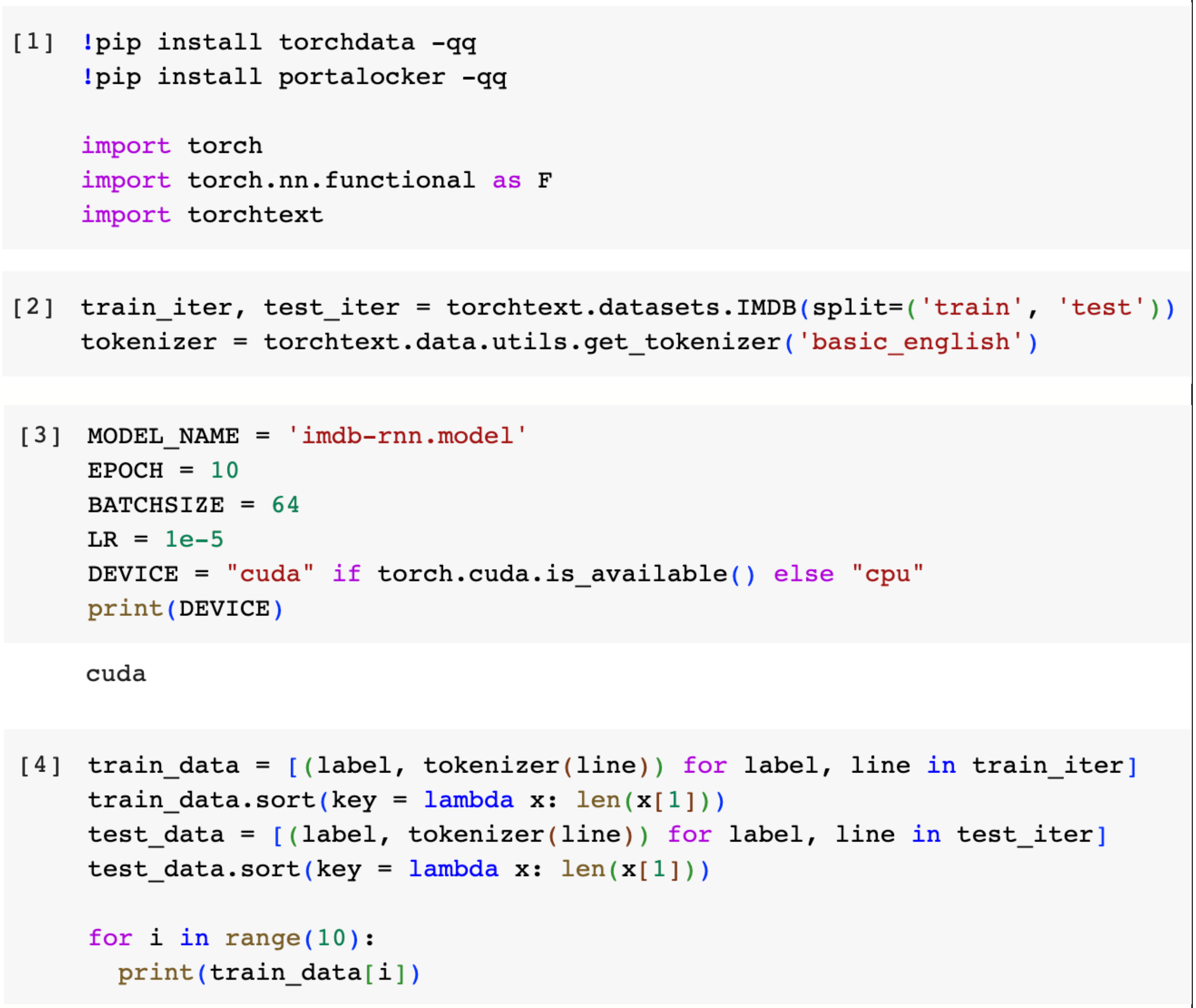
名前(Name): Pham Quoc Cuong

問題 1.

（プログラム）

ライブラリをインストールするためのコマンドです。

`-qq`は、コンソールにログを表示しないフラグです。



```
[1]  !pip install torchdata -qq
     !pip install portalocker -qq

     import torch
     import torch.nn.functional as F
     import torchtext
```

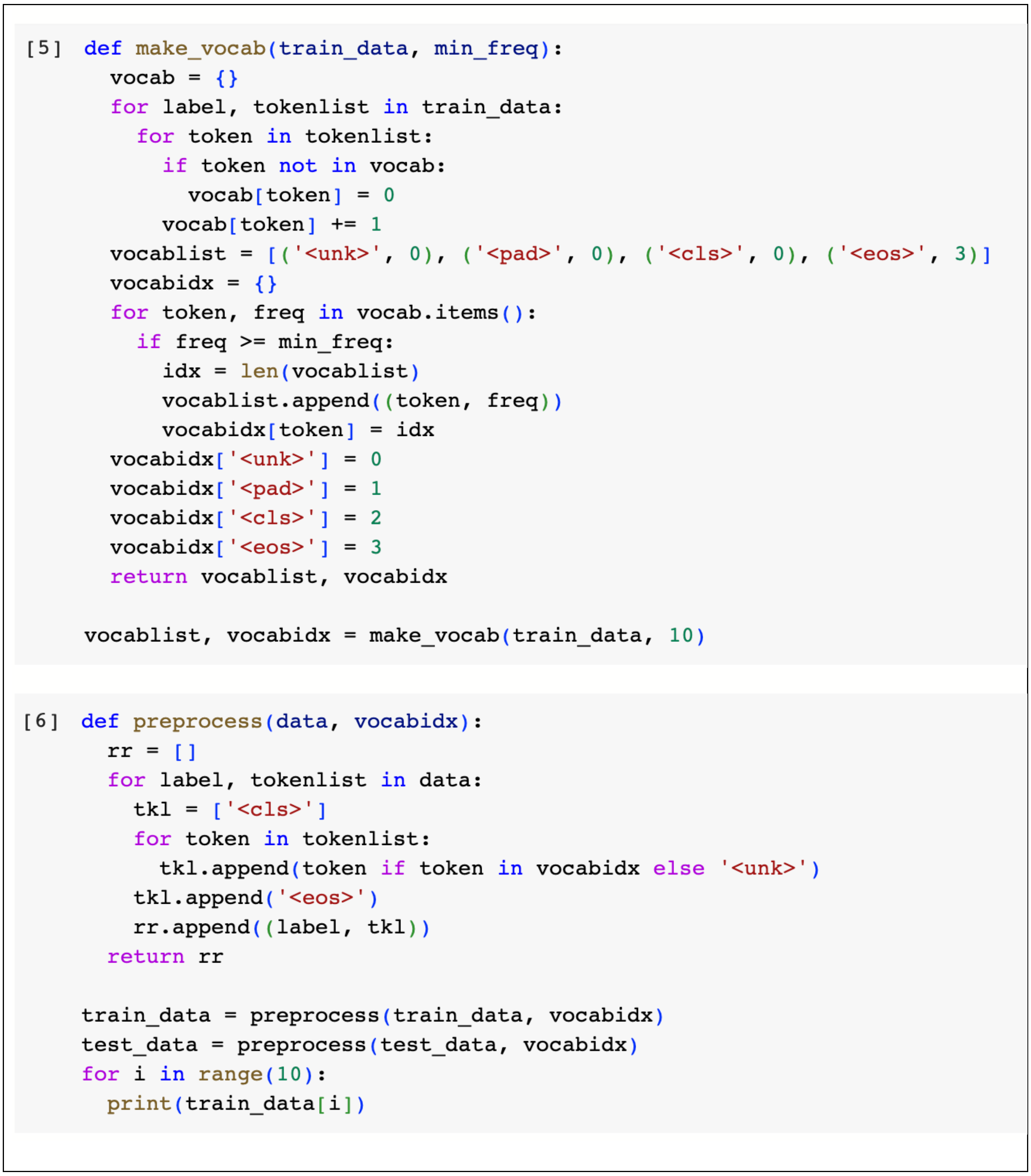
```
[2]  train_iter, test_iter = torchtext.datasets.IMDB(split=('train', 'test'))
     tokenizer = torchtext.data.utils.get_tokenizer('basic_english')
```

```
[3]  MODEL_NAME = 'imdb-rnn.model'
     EPOCH = 10
     BATCHSIZE = 64
     LR = 1e-5
     DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
     print(DEVICE)

     cuda
```

```
[4]  train_data = [(label, tokenizer(line)) for label, line in train_iter]
     train_data.sort(key = lambda x: len(x[1]))
     test_data = [(label, tokenizer(line)) for label, line in test_iter]
     test_data.sort(key = lambda x: len(x[1]))

     for i in range(10):
       print(train_data[i])
```



[5]

```
def make_vocab(train_data, min_freq):
  vocab = {}
  for label, tokenlist in train_data:
    for token in tokenlist:
      if token not in vocab:
        vocab[token] = 0
      vocab[token] += 1
  vocablist = [('<unk>', 0), ('<pad>', 0), ('<cls>', 0), ('<eos>', 3)]
  vocabidx = {}
  for token, freq in vocab.items():
    if freq >= min_freq:
      idx = len(vocablist)
      vocablist.append((token, freq))
      vocabidx[token] = idx
  vocabidx['<unk>'] = 0
  vocabidx['<pad>'] = 1
  vocabidx['<cls>'] = 2
  vocabidx['<eos>'] = 3
  return vocablist, vocabidx

vocablist, vocabidx = make_vocab(train_data, 10)
```

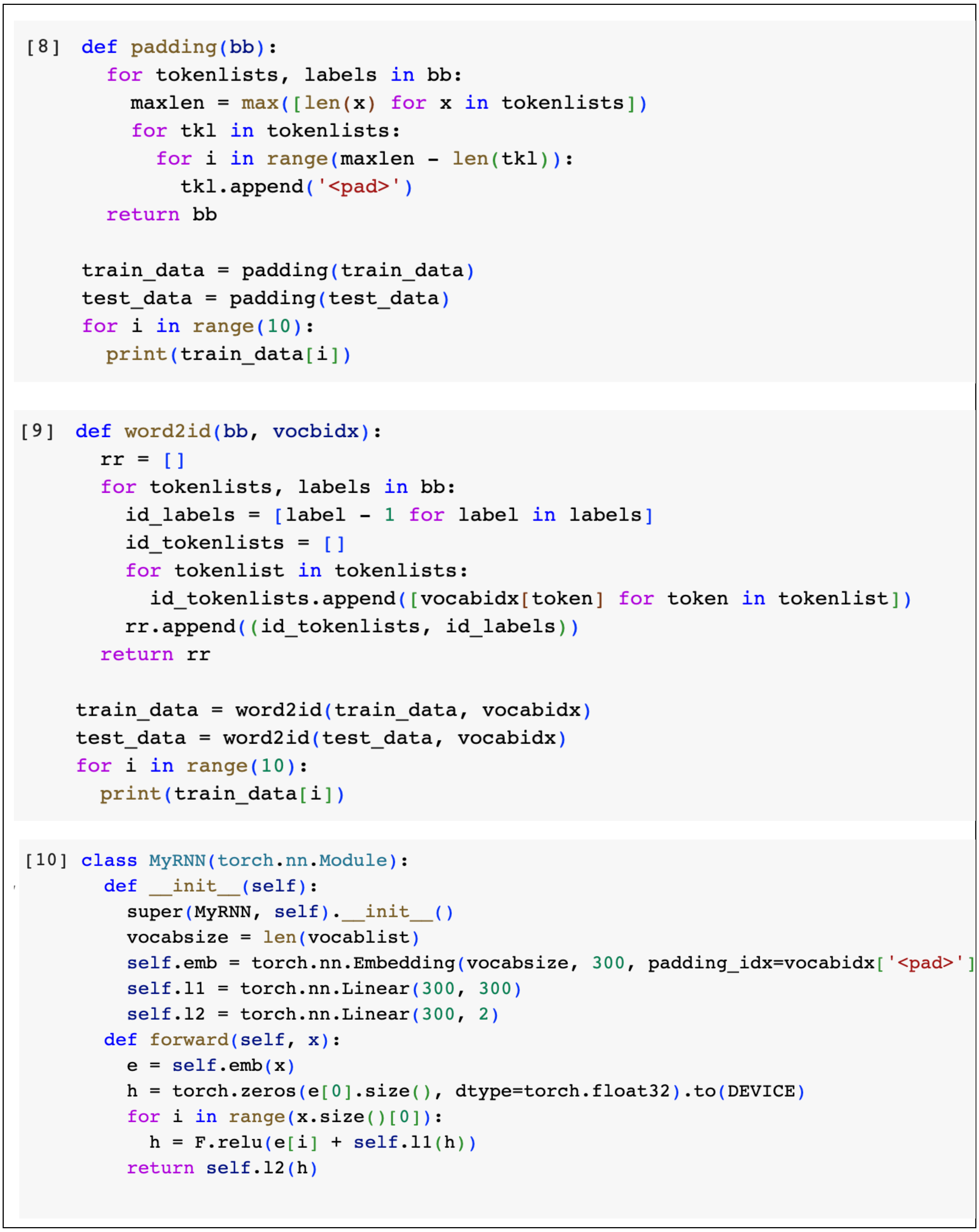
[6]

```
def preprocess(data, vocabidx):
  rr = []
  for label, tokenlist in data:
    tkl = ['<cls>']
    for token in tokenlist:
      tkl.append(token if token in vocabidx else '<unk>')
    tkl.append('<eos>')
    rr.append((label, tkl))
  return rr

train_data = preprocess(train_data, vocabidx)
test_data = preprocess(test_data, vocabidx)
for i in range(10):
  print(train_data[i])
```

```
[7]  def make_batch(data, batchsize):
       bb = []
       blabel = []
       btokenlist = []
       for label, tokenlist in data:
         blabel.append(label)
         btokenlist.append(tokenlist)
         if len(blabel) >= batchsize:
           bb.append((btokenlist, blabel))
           blabel = []
           btokenlist = []
       if len(blabel) > 0:
         bb.append((btokenlist, blabel))
       return bb

     train_data = make_batch(train_data, BATCHSIZE)
     test_data = make_batch(test_data, BATCHSIZE)
     for i in range(10):
       print(train_data[i])
```



```
[8]  def padding(bb):
       for tokenlists, labels in bb:
         maxlen = max([len(x) for x in tokenlists])
         for tkl in tokenlists:
           for i in range(maxlen - len(tkl)):
             tkl.append('<pad>')
       return bb

     train_data = padding(train_data)
     test_data = padding(test_data)
     for i in range(10):
       print(train_data[i])
```

```
[9]  def word2id(bb, vocbidx):
       rr = []
       for tokenlists, labels in bb:
         id_labels = [label - 1 for label in labels]
         id_tokenlists = []
         for tokenlist in tokenlists:
           id_tokenlists.append([vocabidx[token] for token in tokenlist])
         rr.append((id_tokenlists, id_labels))
       return rr

     train_data = word2id(train_data, vocabidx)
     test_data = word2id(test_data, vocabidx)
     for i in range(10):
       print(train_data[i])
```

```
[10] class MyRNN(torch.nn.Module):
       def __init__(self):
         super(MyRNN, self).__init__()
         vocabsize = len(vocablist)
         self.emb = torch.nn.Embedding(vocabsize, 300, padding_idx=vocabidx['<pad>']
         self.l1 = torch.nn.Linear(300, 300)
         self.l2 = torch.nn.Linear(300, 2)
       def forward(self, x):
         e = self.emb(x)
         h = torch.zeros(e[0].size(), dtype=torch.float32).to(DEVICE)
         for i in range(x.size()[0]):
           h = F.relu(e[i] + self.l1(h))
         return self.l2(h)
```

```
[11] def train():
       model = MyRNN().to(DEVICE)
       optimizer = torch.optim.Adam(model.parameters(), lr = LR)
       for epoch in range(EPOCH):
         loss = 0
         for tokenlists, labels in train_data:
           tokenlists = torch.tensor(tokenlists, dtype=torch.int64).transpose(0,1).to(DEVICE)
           labels = torch.tensor(labels, dtype=torch.int64).to(DEVICE)
           optimizer.zero_grad()
           y = model(tokenlists)
           batchloss = F.cross_entropy(y, labels)
           batchloss.backward()
           optimizer.step()
           loss = loss + batchloss.item()

         print("epoch", epoch, ": loss", loss)
       torch.save(model.state_dict(), MODEL_NAME)
```

```
[12] def test():
       total = 0
       correct = 0
       model = MyRNN().to(DEVICE)
       model.load_state_dict(torch.load(MODEL_NAME))
       model.eval()
       for tokenlists, labels in test_data:
         total += len(labels)
         tokenlists = torch.tensor(tokenlists, dtype=torch.int64).transpose(0,1).to(DEVICE)
         labels = torch.tensor(labels, dtype=torch.int64).to(DEVICE)
         y = model(tokenlists)
         pred_labels = y.max(dim=1)[1]
         correct += (pred_labels == labels).sum()

       print("correct:", correct.item())
       print("total:", total)
       print("accuracy:", (correct.item()/float(total)))
```

(実行結果)

データセット内のレビュー文を表示する

```
(1, ['this', 'movie', 'is', 'terrible', 'but', 'it', 'has', 'some', 'good', 'effects', '.'])
(1, ['i', 'wouldn', "'", 't', 'rent', 'this', 'one', 'even', 'on', 'dollar', 'rental', 'night', '.'])
(1, ['ming', 'the', 'merciless', 'does', 'a', 'little', 'bardwork', 'and', 'a', 'movie', 'most', 'foul', '!'])
(2, ['adrian', 'pasdar', 'is', 'excellent', 'is', 'this', 'film', '.', 'he', 'makes', 'a', 'fascinating', 'woman', '.'])
(1, ['you', "'", 'd', 'better', 'choose', 'paul', 'verhoeven', "'", 's', 'even', 'if', 'you', 'have', 'watched', 'it', '.'])
(1, ['long', ',', 'boring', ',', 'blasphemous', '.', 'never', 'have', 'i', 'been', 'so', 'glad', 'to', 'see', 'ending', 'credits', '
(1, ['a', 'rating', 'of', '1', 'does', 'not', 'begin', 'to', 'express', 'how', 'dull', ',', 'depressing', 'and', 'relentlessly', 'ba
(2, ['this', 'is', 'the', 'definitive', 'movie', 'version', 'of', 'hamlet', '.', 'branagh', 'cuts', 'nothing', ',', 'but', 'there',
(2, ['i', 'don', "'", 't', 'know', 'why', 'i', 'like', 'this', 'movie', 'so', 'well', ',', 'but', 'i', 'never', 'get', 'tired', 'of'
(1, ['the', 'characters', 'are', 'unlikeable', 'and', 'the', 'script', 'is', 'awful', '.', 'it', "'", 's', 'a', 'waste', 'of', 'the'
```

文の先頭に"<cls>"を追加し、文の末尾に"<eos>"を追加し、語彙に存在しない単語の場合には"<unk>"と表示される場合、データセット内のレビュー文を出力してください。

```
(1, ['<cls>', 'this', 'movie', 'is', 'terrible', 'but', 'it', 'has', 'some', 'good', 'effects', '.', '<eos>'])
(1, ['<cls>', 'i', 'wouldn', "'", 't', 'rent', 'this', 'one', 'even', 'on', 'dollar', 'rental', 'night', '.', '<eos>'])
(1, ['<cls>', 'ming', 'the', 'merciless', 'does', 'a', 'little', '<unk>', 'and', 'a', 'movie', 'most', 'foul', '!', '<eos>'])
(2, ['<cls>', 'adrian', 'pasdar', 'is', 'excellent', 'is', 'this', 'film', '.', 'he', 'makes', 'a', 'fascinating', 'woman', '.', '<eos>']
(1, ['<cls>', 'you', "'", 'd', 'better', 'choose', 'paul', 'verhoeven', "'", 's', 'even', 'if', 'you', 'have', 'watched', 'it', '.', '<eo
(1, ['<cls>', 'long', ',', 'boring', ',', 'blasphemous', '.', 'never', 'have', 'i', 'been', 'so', 'glad', 'to', 'see', 'ending', 'credits
(1, ['<cls>', 'a', 'rating', 'of', '1', 'does', 'not', 'begin', 'to', 'express', 'how', 'dull', ',', 'depressing', 'and', 'relentlessly',
(2, ['<cls>', 'this', 'is', 'the', 'definitive', 'movie', 'version', 'of', 'hamlet', '.', 'branagh', 'cuts', 'nothing', ',', 'but', 'ther
(2, ['<cls>', 'i', 'don', "'", 't', 'know', 'why', 'i', 'like', 'this', 'movie', 'so', 'well', ',', 'but', 'i', 'never', 'get', 'tired',
(1, ['<cls>', 'the', 'characters', 'are', 'unlikeable', 'and', 'the', 'script', 'is', 'awful', '.', 'it', "'", 's', 'a', 'waste', 'of', '
```

文のデータをトレーニングのために複数のバッチに分割してください。

```
([['<cls>', 'this', 'movie', 'is', 'terrible', 'but', 'it', 'has', 'some', 'good', 'effects', '.', '<eos>'], ['<cls>', 'i', 'wouldn',
([['<cls>', 'the', 'way', 'this', 'story', 'played', 'out', 'and', 'the', 'interaction', 'between', 'the', '2', 'lead', 'characters',
([['<cls>', 'i', 'have', 'never', 'seen', 'such', 'terrible', 'performances', 'in', 'all', 'my', 'life', '.', 'everyone', 'in', 'the'
([['<cls>', 'i', 'really', 'liked', 'this', 'movie', 'despite', 'one', 'scene', 'that', 'was', 'pretty', 'bad', '(', 'the', 'one', 'w
([['<cls>', 'the', 'only', 'kung', 'fu', 'epic', 'worth', 'watching', '.', 'the', 'best', 'training', 'ever', '.', 'the', 'main', 'ch
([['<cls>', 'i', 'didn', "'", 't', 'mind', 'the', 'film', 'that', 'much', ',', 'but', 'it', 'was', 'incredibly', 'dull', 'and', 'bori
([['<cls>', 'this', 'is', 'the', 'definite', 'lars', 'von', 'trier', 'movie', ',', 'my', 'favorite', ',', 'i', 'rank', 'it', 'higher'
([['<cls>', 'i', 'saw', 'this', 'movie', 'on', 'mystery', 'science', 'theater', '300', '.', 'it', 'sucked', 'so', 'much', '.', 'if',
([['<cls>', 'a', 'timeless', 'classic', ',', 'wonderfully', 'acted', 'with', 'perfect', 'location', 'settings', ',', '<unk>', 'a', 'm
([['<cls>', 'this', 'version', 'of', 'the', 'charles', 'dickens', 'novel', 'features', 'george', 'c', 'scott', 'as', 'the', 'scrooge'
```

データセット内の文の最大長に基づいて、文の末尾にパディングを追加します。

```
['<cls>', 'this', 'movie', 'is', 'terrible', 'but', 'it', 'has', 'some', 'good', 'effects', '.', '<eos>', '<pad>', '<pad>', '<pad>', '<p
['<cls>', 'the', 'way', 'this', 'story', 'played', 'out', 'and', 'the', 'interaction', 'between', 'the', '2', 'lead', 'characters', 'may
['<cls>', 'i', 'have', 'never', 'seen', 'such', 'terrible', 'performances', 'in', 'all', 'my', 'life', '.', 'everyone', 'in', 'the', 'en
['<cls>', 'i', 'really', 'liked', 'this', 'movie', 'despite', 'one', 'scene', 'that', 'was', 'pretty', 'bad', '(', 'the', 'one', 'when',
['<cls>', 'the', 'only', 'kung', 'fu', 'epic', 'worth', 'watching', '.', 'the', 'best', 'training', 'ever', '.', 'the', 'main', 'charact
['<cls>', 'i', 'didn', "'", 't', 'mind', 'the', 'film', 'that', 'much', ',', 'but', 'it', 'was', 'incredibly', 'dull', 'and', 'boring',
['<cls>', 'this', 'is', 'the', 'definite', 'lars', 'von', 'trier', 'movie', ',', 'my', 'favorite', ',', 'i', 'rank', 'it', 'higher', 'th
['<cls>', 'i', 'saw', 'this', 'movie', 'on', 'mystery', 'science', 'theater', '300', '.', 'it', 'sucked', 'so', 'much', '.', 'if', 'i',
['<cls>', 'a', 'timeless', 'classic', ',', 'wonderfully', 'acted', 'with', 'perfect', 'location', 'settings', ',', '<unk>', 'a', 'marvel
['<cls>', 'this', 'version', 'of', 'the', 'charles', 'dickens', 'novel', 'features', 'george', 'c', 'scott', 'as', 'the', 'scrooge', '.'
```

ボキャブラリーインデックスに基づいて、単語を数字に変換してトレーニングする。

```
{'this': 4, 'movie': 5, 'is': 6, 'terrible': 7, 'but': 8, 'it': 9, 'has': 10, 'some': 11, 'good': 12, 'effects': 13, '.': 14, 'i': 15
```

```
([[2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [2, 15,
([[2, 27, 424, 4, 156, 681, 369, 32, 27, 682, 662, 27, 683, 684, 97, 685, 684, 332, 62, 544, 138, 51, 27, 686, 687, 242, 688, 32, 689,
([[2, 15, 52, 58, 245, 429, 7, 193, 158, 148, 250, 1034, 14, 694, 158, 27, 1242, 39, 172, 1243, 1244, 14, 70, 20, 1147, 1245, 158, 27,
([[2, 15, 405, 542, 4, 5, 334, 20, 362, 138, 172, 648, 77, 358, 27, 20, 135, 1729, 32, 1730, 85, 1731, 158, 27, 1732, 591, 14, 27, 156,
([[2, 27, 323, 2166, 588, 2167, 160, 96, 14, 27, 359, 2168, 173, 14, 27, 1148, 264, 2169, 30, 1060, 733, 17, 50, 22, 539, 2170, 2171, 2
([[2, 15, 381, 17, 18, 777, 27, 39, 138, 266, 55, 8, 9, 172, 1258, 74, 32, 56, 14, 30, 453, 1297, 496, 32, 84, 8, 83, 62, 118, 2277, 11
([[2, 4, 6, 27, 397, 1703, 2741, 2742, 5, 55, 250, 293, 55, 15, 2924, 9, 1313, 206, 2925, 27, 2926, 144, 27, 2927, 2928, 158, 27, 813,
([[2, 15, 353, 4, 5, 22, 1412, 3251, 768, 3252, 14, 9, 2486, 60, 266, 14, 51, 15, 3253, 17, 18, 59, 96, 9, 22, 3254, 55, 15, 434, 376,
([[2, 30, 3523, 179, 55, 3524, 514, 120, 799, 2209, 3525, 55, 0, 30, 2495, 3526, 5, 14, 30, 972, 156, 0, 120, 531, 32, 1818, 14, 15, 75
([[2, 4, 79, 68, 27, 2987, 3851, 3705, 827, 3852, 2782, 2479, 180, 27, 3853, 14, 1185, 3854, 55, 128, 27, 3384, 68, 2479, 55, 183, 329,
train()
```

```
epoch 0 : loss 245.0376599431038
epoch 1 : loss 238.89905858039856
epoch 2 : loss 237.6389603316784
epoch 3 : loss 236.95001085102558
epoch 4 : loss 236.46642750501633
epoch 5 : loss 236.08276760578156
epoch 6 : loss 235.7479064911604
epoch 7 : loss 235.44679905474186
epoch 8 : loss 235.15076033771038
epoch 9 : loss 234.8589846342802
```

```
test()
```

```
correct: 16887
total: 25000
accuracy: 0.67548
```