第四次作业

设计文字处理的类

代码部分

import jieba  
  
class Tokenizer:  
 def \_\_init\_\_(self, chars, coding = 'c', PAD = 0):  
 # 输入将要需要操作的文本（一个字符串的列表），  
 #这里需要完成词典的构建（即汉字到正整数的唯一映射的确定）。注意构建词典一是要根据  
 #coding来选择按词构建（coding='w')，还是按字构建，默认按字构建；PAD默认为0。  
 #global match\_dict,sentence\_list,mode  
 self.match\_dict = {'[PAD]': PAD}  
 self.sentence\_list = chars  
 self.mode = coding  
 code = 1  
 if coding == 'c':  
 for sentence in chars:  
 for word in sentence:  
 if word not in self.match\_dict:  
 self.match\_dict[word] = code  
 code += 1  
 elif coding == 'w':  
 for sentence in chars:  
 words = jieba.lcut(sentence)  
 for word in words:  
 if word not in self.match\_dict:  
 self.match\_dict[word] = code  
 code += 1  
 #print(match\_dict)  
 def tokenize(self, sentence):  
 #输入一句话，返回分词(字）后的字符列表(list\_of\_chars)。  
 if self.mode == 'c':  
 list\_of\_chars = list(sentence)  
 else:  
 list\_of\_chars = jieba.lcut(sentence)  
 #print(list\_of\_chars)  
 return list\_of\_chars  
 def encode(self, list\_of\_chars):  
 #输入字符(字或者词）的字符列表，返回转换后的数字列表(tokens)  
 tokens = []  
 for word in list\_of\_chars:  
 tokens.append(self.match\_dict[word])  
 return tokens  
 def trim(self, tokens, seq\_len):  
 #输入数字列表tokens，整理数字列表的长度。不足seq\_len的  
 #部分用PAD补足，超过的部分截断。  
 if len(tokens) >= seq\_len:  
 tokens = tokens[:seq\_len]  
 else:  
 end = [0 for i in range(seq\_len-len(tokens))]  
 tokens.extend(end)  
 return tokens  
 def decode(self, tokens):  
 #将模型输出的数字列表翻译回句子。如果有PAD，输出'[PAD]'。  
 sentence = ''  
 for i in tokens:  
 for key in self.match\_dict:  
 if self.match\_dict[key] == i:  
 sentence+=key  
 return sentence  
 def get(self, seq\_len):  
 #返回所有文本（chars)的长度为seq\_len的tokens。  
 for sentence in self.sentence\_list:  
 if len(sentence) == seq\_len:  
 self.encode(self.tokenize(sentence))

测试部分

if \_\_name\_\_ == '\_\_main\_\_':  
 with open(r'jd\_comments.txt','r',encoding='utf-8') as f:  
 text\_line = f.read()  
 text\_list = text\_line.split('\n')  
 t1 = Tokenizer(text\_list, coding='c', PAD = 0 )  
 list\_of\_chars = t1.tokenize(text\_list[0])  
 print(list\_of\_chars)  
 tokens = t1.encode(list\_of\_chars)  
 print(tokens)  
 tokens = t1.trim(tokens, 10)  
 print(tokens)  
 sentence = t1.decode(tokens)  
 print(sentence)  
 tokens = t1.get(10)  
 print(tokens)

测试结果

['帮', '家', '里', '老', '人', '选', '购', '的', '，', '对', '主', '机', '外', '观', '很', '满', '意', '，', '开', '机', '速', '度', '也', '不', '错', '，', '运', '行', '声', '音', '很', '小', '，', '暂', '时', '没', '发', '现', '什', '么', '问', '题', '，', '希', '望', '可', '以', '用', '的', '住', '。']

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 9, 18, 12, 19, 20, 21, 22, 23, 9, 24, 25, 26, 27, 15, 28, 9, 29, 30, 31, 32, 33, 34, 35, 36, 37, 9, 38, 39, 40, 41, 42, 8, 43, 44]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

帮家里老人选购的，对

None