文本处理的Tokenizer类

STEP 1 初始化函数

创建词典 self.dictionary

```python
class Tokenizer:
    def __init__(self, strings, coding, PAD):
        if coding == "c":
            word_list=list(set(list("".join(strings)))) #按照字切分文本
        else:
            word_list=list(set(jieba.lcut("".join(strings)))) #按照 jieba 分词切分文本
        order = [i for i in range(len(word_list))]
        self.dictionary = dict(zip(word_list, order))
```

Python

简单测试：

若 coding="w":{'我': 0, '高兴': 1, '今天': 2, '很': 3, '开心': 4}

若 coding="c":{'心': 0, '开': 1, '很': 2, '兴': 3, '今': 4, '我': 5, '天': 6, '高': 7}

STEP 2 分词

将字符串划分成列表。

```python
    def tokenize(self,sentence):
        if self.coding == "c":
            list_of_chars = list(sentence)
        else:
            list_of_chars = jieba.lcut(sentence)
        return list_of_chars
```

Python

简单测试：

若 coding="w":["我","开心"]

若 coding="c":["我","开","心"]

STEP 3 编码

将列表转换成数字列表。

```python
    def encode(self,list_of_chars):
        tokens = []
        for i in list_of_chars:
            tokens.append(self.dictionary[i])
        return tokens
```

Python

简单测试：

若 coding="w":[3, 1]

若 coding="c":[5, 1, 7]

STEP 4 补全

将不足 seq_len 的长度补齐。

```python
    def trim(self,tokens,seq_len):
        delta = seq_len-len(tokens)
```

```python
        if delta > 0:
            for i in range(delta):
                tokens.append(0)
        return tokens
```

简单测试：

若 coding="w":[3, 1, 0, 0, 0]

若 coding="c":[5, 1, 7, 0, 0]

STEP 5 解码

把数字列表还原成字符。

```python
    def decode(self,tokens):
        new_dict = {v : k for k, v in self.dictionary.items()} #将原字典的键和值对换
        sentence_list = []
        for i in tokens:
            sentence_list.append(new_dict[i])
        return sentence_list
```

简单测试：

若 coding="w":['我', '开心', 'PAD', 'PAD', 'PAD']

若 coding="c":['我', '开', '心', 'PAD', 'PAD']

STEP 6 全部编码

把长度刚好等于 seq_len 的编码列表输出。

```python
    def encode_all(self,seq_len):
        all_tokens = []
        if self.coding == "c":
            for i in self.chars:
                if len(list(i)) == seq_len:
                    all_tokens.append(self.encode(self.tokenize(i)))
        else:
            for i in self.chars:
                if len(jieba.lcut(i)) == seq_len:
                    all_tokens.append(self.encode(self.tokenize(i)))
        return all_tokens
```

简单测试：

若 coding="w":[[3, 5, 4, 1], [3, 5, 4, 2]]

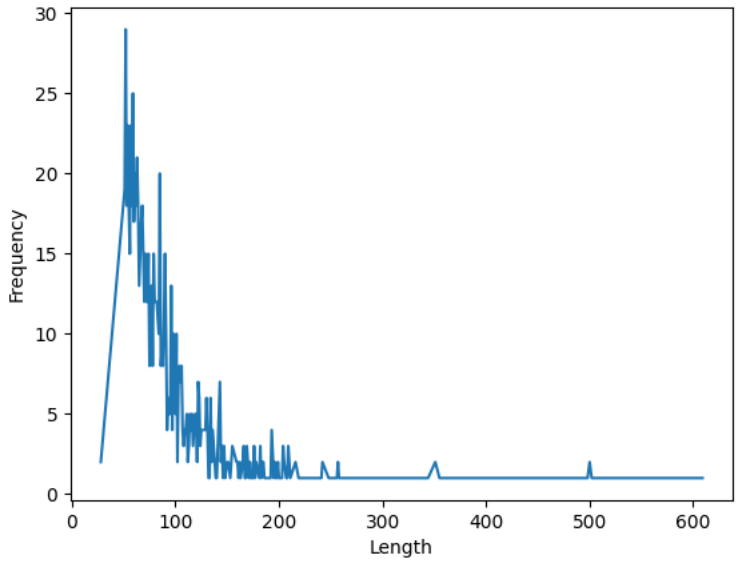
若 coding="c":[[2, 6, 1, 8, 4, 5], [2, 6, 1, 8, 3, 7]]

STEP 7 确认合适的 seq_len

对电商评论进行分词，确定评论的长度分布。

```python
if __name__ == "__main__":
    sentence_list = []
    with open('./jd_comments.txt', 'r',encoding='utf-8') as f:
        for line in f.readlines():
            sentence_list.append(line.strip())
    tokenizer = Tokenizer(sentence_list, "c", 0)
    len_list = []
    for i in tokenizer.chars:
        len_list.append(len(tokenizer.encode(tokenizer.tokenize(i))))
    len_list = sorted(len_list)
    frequency_dict = {}
    for i in len_list:
        frequency_dict[i] = frequency_dict.get(i,0) + 1
    x = list(frequency_dict.keys())
    y = list(frequency_dict.values())
```

```python
    plt.plot(x,y)
    plt.xlabel("Length")
    plt.ylabel("Frequency")
    plt.show()
```



由图可知，seq_len 设定为 100 左右比较合适。