The guideline of Takin

数据清洗函数

1. delete_escape_character：删除转义字符

```
print(takin.delete_escape_character("这些曾经\n我们\t都好\n那些事情\r", lang="zh"))
print(takin.delete_escape_character("这些曾经。\n我们\t都好。\n那些事情\r", lang="zh"))
print(takin.delete_escape_character("Today is sunday\nwe are \thappy.", lang="en"))
print(takin.delete_escape_character("Today is sunday.\nwe are \thappy.", lang="en"))

这些曾经。我们都好。那些事情
这些曾经。我们都好。那些事情
Today is sunday. we are happy.
Today is sunday. we are happy.
```

2. delete_extra_whitespace: 删除文本中的多余空白

```
print(takin.delete_extra_whitespace("我 们  都非   常快 乐   。", lang="zh"))
print(takin.delete_extra_whitespace("Takin  ,    is very   useful  .    ", lang="en"))

我们都非常快乐。
Takin, is very useful.
```

3. delete_digit: 删除文本中的数字

```
print(takin.delete_digit("今天6天777气7908762345不错！"))

今天天气不错！
```

4. delete_punctuation: 删除文本中的所有标点符号

```
print(takin.delete_punctuation("Long;:,.\"??!''·！？；，。：“”‘’《》[『』]〈~ηe®·•~~#/*&$|★▶><\^@+[=]() 〈) {%_}?…]"))

Long
```

5. delete_letter: 删除所有字母

```
print(takin.delete_letter("明天将会是一个beautiful晴朗的天气"))

明天将会是一个晴朗的天气
```

6. delete_chinese: 删除所有汉字

```
print(takin.delete_chinese("This is another 胜利victory!"))

This is another victory!
```

7. delete_bracket: 删除括号及括号里的内容

```
print(takin.delete_bracket("机器阅读理解（MRC），【旨在】教机器理解人类语言(language){热爱学习}[hah]"))

机器阅读理解，教机器理解人类语言
```

8. delete_series_number: 删除序号

```
print(takin.delete_series_number("1.努力工作；(2).用心学习；(3)锻炼身体；4).热爱家庭 5。快乐；6)学习, 7)、（一）、集中学习 （十五）高度集中 （一百二十

努力工作；用心学习；锻炼身体；热爱家庭 快乐；学习, 集中学习 高度集中
```

9. delete_repeated_punc: 连续重复的标点符号只保留一次

```
print(takin.delete_repeated_punc("what's up??????????????...◦◦《《《"))

what's up?.◦《
```

数据划分函数

1. split_dataset: 给定一个原始数据集，按照比例将其划分为训练集、验证集、测试集

```
corpus = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J"]
train, dev, test = takin.split_dataset(corpus, "7:2:1", is_shuffle=False)
print(len(train), train)
print(len(dev), dev)
print(len(test), test)

7 ['A', 'B', 'C', 'D', 'E', 'F', 'G']
2 ['H', 'I']
1 ['J']
```

```
corpus = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J"]
train, test = takin.split_dataset(corpus, "7:3", is_shuffle=False)
print(len(train), train)
print(len(test), test)

7 ['A', 'B', 'C', 'D', 'E', 'F', 'G']
3 ['H', 'I', 'J']
```

2. split_dataset_by_class: corpus中每个元素是dict，按照类别进行数据切分

```
data = []

with open("./data/train.txt", "r", encoding="utf-8") as f:
    for line in f:
        ele = line.strip().split("\t")
        data.append({"text": ele[1], "label": ele[0]})

with open("./data/test.txt", "r", encoding="utf-8") as f:
    for line in f:
        ele = line.strip().split("\t")
        data.append({"text": ele[1], "label": ele[0]})

print(len(data))

train, dev, test = takin.split_dataset_by_class(data, "7:2:1", cate="label", is_shuffle=True)
print(len(train))
print(len(dev))
print(len(test))

print(test[-5:])

7028
3类别的数据样例为：725
5类别的数据样例为：1352
4类别的数据样例为：2447
1类别的数据样例为：1372
0类别的数据样例为：702
```

```
2类别的数据样例为：430
4922
1408
698
[{'text': 'xxxxxxxxxxxxxxxxxxx工商银行户名赵英杰打到这张卡上', 'label': '5'}, {'text': 'xxxxxxxxxxxxxxxxxxx用户名:赵坤坤建行卡号', 'label': '5
```

数据解析函数

```
print(takin.read_txt("./resources/parsing_examples/test.txt"))
print(takin.read_docx("./resources/parsing_examples/test.docx"))
print(takin.read_pptx("./resources/parsing_examples/test.pptx"))
print(takin.read_pdf("./resources/parsing_examples/test.pdf"))
print(takin.read_html("./resources/parsing_examples/test.html"))
print(takin.read_eml("./resources/parsing_examples/test.eml"))
```