【报修开发】缝针缝线市一南院 - Task #20280

OCR 大小写，空格横线，相近字原优化

2025-01-10 09:45 - 尹駟（Antoine）

状态:	Feedback	**开始日期:**	2025-01-10
优先级:	Normal	**计划完成日期:**	
指派给:	王雷（Jerry）	**% 完成:**	0%
类别:		**预期时间:**	0.00 小时
目标版本:		**耗时:**	16.50 小时
客户名称:		**现场FSW统计:**	否
关键字:		**产品名称:**	缝针缝线柜

描述

ocr识别时存在识别异常问题，如大大小写异常，相近字符识别错误等，针对这些异常
主要针对领用时ocr识别中的批次，效期信息进行处理
处理方案为，通过预处理字符串的方式实现
1，增加字符串预预处理配置，规则如下图所示 {
"batchNumber": [{
"source": "5",
"target": "S"
},{
"source": "1",
"target": "I"
}],
"expireationDate": [{
"source": "-",
"target": ""
},{
"source": " ",
"target": ""
}],
}
客户端进行匹配时，将ocr识别的结果按照以上规则进行预处理，如识别结果时“ OC55
U ”， 金国预处理后的结果是“ OCSSU ”，同时将库存中的批次，效期信息分别按以上规则进行预处理，再进行匹配，
问题点
1，客户端需要对库存内的货品批次效期进行预处理，这个可以在拉库存时做掉，将值保存在客户端
这两个值不放在服务端是为防止配置文件发生编号导致老的数据无效，其次是服务端和客户分别处理可莪能由于配置文件不一致或实现方法不一致导致结果有差异
1，两个不同批次的货品，进行预处理之后的值可能是一样，导致无法区分，
理论上不同批次，同一效期概率较低，这种情况待发生之后再处理

参考：

在处理字符串比较时，如果需要忽略部分字符的差异（例如忽略大小写、空格、标点符号等），可以通过以下方法实现：

---

1. **1. 预处理字符串**
   在比较之前，对字符串进行预处理，去除或标准化需要忽略的字符。

1. **示例：忽略大小写和空格**
   ```python
   def preprocess_string(s):  # 转换为小写并去除空格
   return s.lower().replace(" ", "")

str1 = "Hello World"
str2 = "hello world"

1. 预处理后比较
   if preprocess_string(str1) == preprocess_string(str2):
   print("字符串相同（忽略大小写和空格）")
   else:
   print("字符串不同")
   ```

```

**输出**：
```
字符串相同（忽略大小写和空格）
```

---

1. **使用正则表达式**
   通过正则表达式去除或替换需要忽略的字符。

   1. **示例：忽略标点符号**
      ```python
      import re

def remove_punctuation(s):  # 去除标点符号
return re.sub(r'[^\w\s]', '', s)

str1 = "Hello, World!"
str2 = "Hello World"

   1. 去除标点符号后比较
      if remove_punctuation(str1) == remove_punctuation(str2):
      print("字符串相同（忽略标点符号）")
      else:
      print("字符串不同")
      ```

输出：
```
字符串相同（忽略标点符号）
```

---

1. **自定义比较函数**
   编写一个自定义的比较函数，根据需要忽略的规则进行比较。

   1. **示例：忽略特定字符**
      ```python
      def compare_strings_ignore_chars(str1, str2, ignore_chars):  # 去除需要忽略的字符
      for char in ignore_chars:
      str1 = str1.replace(char, '')
      str2 = str2.replace(char, '')
      return str1 == str2

str1 = "Hello-World"
str2 = "HelloWorld"
ignore_chars = ["-", "_"]

   1. 比较字符串
      if compare_strings_ignore_chars(str1, str2, ignore_chars):
      print("字符串相同（忽略 - 和 _）")
      else:
      print("字符串不同")
      ```

输出：
```
字符串相同（忽略 - 和 _）
```

---

1. **使用模糊匹配库**
   如果需要更灵活的字符串比较，可以使用模糊匹配库（如 `fuzzywuzzy`）。

   1. **安装 `fuzzywuzzy`**：
      ```bash

pip install fuzzywuzzy
```

1. **示例：模糊匹配**
```python
from fuzzywuzzy import fuzz

str1 = "Hello World"
str2 = "hello world"

1. 计算相似度
similarity = fuzz.ratio(str1.lower(), str2.lower())

if similarity > 90: # 设置相似度阈值
print("字符串相似（忽略大小写）")
else:
print("字符串不相似")
```

输出：
```
字符串相似（忽略大小写）
```

---

1. **5. 综合示例**
以下是一个综合示例，结合预处理和模糊匹配：

```python
import re
from fuzzywuzzy import fuzz

def preprocess_string(s):  # 转换为小写并去除标点符号和空格
s = s.lower()
s = re.sub(r'[^\w]', '', s)
return s

str1 = "Hello, World!"
str2 = "hello world"

1. 预处理字符串
str1_processed = preprocess_string(str1)
str2_processed = preprocess_string(str2)

1. 计算相似度
similarity = fuzz.ratio(str1_processed, str2_processed)

if similarity > 90: # 设置相似度阈值
print("字符串相似（忽略大小写、空格和标点符号）")
else:
print("字符串不相似")
```

输出：
```
字符串相似（忽略大小写、空格和标点符号）
```

---

1. **6. 总结**
- **预处理字符串**：通过标准化（如小写转换）和去除特定字符（如空格、标点符号）来忽略差异。
- **正则表达式**：用于灵活地去除或替换字符。
- **自定义比较函数**：根据需求实现特定的比较逻辑。
- **模糊匹配**：适用于需要容忍一定差异的场景。

根据你的具体需求，选择合适的方法来实现字符串比较时忽略部分字符的差异！
```

历史记录

#1 - 2025-02-10 18:01 - 王雷（Jerry）

- 描述 已更新。

#2 - 2025-02-11 09:27 - 王雷（Jerry）

- 状态 从 New 变更为 In Progress

#3 - 2025-02-11 09:28 - 王雷（Jerry）

01-10 2.5H
1，ocr识别时，效期，批次信息支持忽略部分字符设计

#4 - 2025-02-18 09:46 - 王雷（Jerry）

02-17 1H
1，功能review，设计完善

#5 - 2025-02-19 09:35 - 王雷（Jerry）

- 状态 从 In Progress 变更为 Feedback

#6 - 2025-02-19 09:38 - 王雷（Jerry）

02-18 7H
1，Google ocr易错字符识别设计，文档提供
2，客户端操作记录查询设计，文档修改提供

#7 - 2025-02-20 09:47 - 王雷（Jerry）

02-19 3H
1，历史业务数据查询修改，补充相关图片和数据格式细节，

#8 - 2025-03-04 10:04 - 王雷（Jerry）

03-03 1H
1，设计调整，补充部分AI参考内容

#9 - 2025-03-05 10:13 - 王雷（Jerry）

03-04 2H
1，设计修改，完善，完成
2，ocr 图片裁剪问题，待处理