专利机翻检测工具 - 使用文档

专利机翻检测工具，用于辅助发现机翻中的疑似问题，方便进一步处理。

线上地址

包括*检测工具*和*可视化工具*两部分，

- 检测工具需要做安装配置，用于生成待分析的*.json文件
- 可视化工具不需要，仅需拷贝views/目录、待分析的*.json文件

操作流程

*检测工具*和*可视化工具*松耦合，适合多种分析和和任务模式

操作流程分为两部分，

- 利用*检测工具*生成检测结果数据，命名方式都是以`-o`参数指作为前缀，包括以下三种：

1. *_visual/，文件夹形式，是用来可视化分析的数据；
2. *_csv/，文件夹形式，生成可以使用excel打开的csv数据；
3. *_errors.txt，文件形式，生成处理错误的日志数据；

- 利用*可视化工具*views/index.html进行分析

检测工具

`run.py`使用python构建，如果缺少安装包，运行前需要pip install缺少的包。

批处理文件bat使用流程如下：

1. 进入workspace文件夹；
2. 将数据复制进入run_input/，无论是excel还是xml还是txt
3. 双击运行bat文件，每种数据类型有对应
4. 在run_output/查看结果

命令行使用流程如下：

1. 准备待分析的原文XML和译文XML
2. 将待分析的原文XML和译文XML放在同一个文件夹下，比如input_folder/
3. 检查和配置config.ini文件
4. 利用如下命令进行分析:

```
# 如下命令可以直接运行
# 需要提前切换到安装包根目录下，如 PatTransErrorDetect/
# 运行后，将在同级目录下，生成*_visual/文件夹、*_csv文件夹、*_errors.txt
python3 src/run.py -i data/US/ -o results/us -c src/config.ini
python3 src/run.py -i data/JP/ -o results/jp -c src/config.ini
python3 src/run.py -i data/CA/ -o results/ca -c src/config.ini
```

```
python3 src/run.py -i data/AT/ -o results/at -c src/config.ini

# excel版的运行方式
python3 src/run.excel.py -i data/RUexcel/ -o results/rucsv -c
src/config.ini

# txt版的运行方式
python3 src/run.txt.py -i data/KRtxt/ -o results/krtxt -c src/config.ini
```

全部的参数列表是：

```
usage: run.py [-h] [-i INPUT_FOLDER] [-o OUTPUT_FOLDER] [-c CONFIG] [-j
JOBS]

Data process tool.

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT_FOLDER, --input_folder INPUT_FOLDER
                        输入文件夹名
  -o OUTPUT_FOLDER, --output_folder OUTPUT_FOLDER
                        输出文件夹名称前缀
  -c CONFIG, --config CONFIG
                        配置文件
  -j JOBS, --jobs JOBS  进程数，一般默认即可
```

接下来详细说一下输入和输出

input_folder

这个文件夹存放了原文XML和译文XML，也就是原文和译文文件同时需要放在相同目录下。

run.py将从文件夹中找到原文XML，并对应找到译文。

原文和译文的对应方式是：

译文的文件名字是在原文文件名的基础上，在.XML之前拼接_trans作为译文的名字，比如：

```
data/JP/2014/JP102014000263706JP00020161241010AFULJA20160711JP005.XML
# 该文件的译文路径为：
#
data/JP/2014/JP102014000263706JP00020161241010AFULJA20160711JP005_trans.XM
L

# 示例操作命令：
python run.py -c config.ini -i data/JP/ -o jp
```

配置文件说明

`config.ini`包含了一些基本设置，其中，DEFAULT是必填的。

DEFAULT配置项

1. `TAG`: 用于指定需要匹配的XML标签
2. `MAX_ERROR_TIMES_PERTAG_PERTYPE`: 在可视化中输出的每个类型的错误数，默认是10
3. `MAX_ERROR_TIMES_PERTAG_PERTYPE_CSV`: 在csv中输出的每个类型的错误数，默认是100
4. `MAX_LENGTH_PERTAG`: 标签内容的最大长度，默认是10240，就是说每个标签下最多有一万字
5. `FILE_NAME_PATTERN`: 原文XML的文件名形式，一般不需要修改

锚点正则配置项

剩下的选项是设置翻译锚点，主要是锚点正则表达式们，格式如下：

```
[名称]
mode=匹配方式
stat=统计方式
escape=排除规则，允许多个
pattern=识别规则，允许多个
```

示例：

```
[HTML字符实体]
__pattern__html_entities_chunk=(&[\w;]+;)
__escape__1=(\&amp\;)
mode=chunk
stat=poly
```

该示例中，设置了名字为“HTML字符实体”的翻译锚点，有一个匹配规则，匹配模式是chunk，统计方式是poly

匹配模式一共有三种：`single`, `chunk`, `multichunk`：

- `single`是指单独匹配，适用于单个字符形式，比如特殊数学符号
- `chunk`是指块匹配，适用于多个字符的字符串形式，比如html实体名称：`&`
- `multichunk`是指多块匹配，适用于复杂的字符串形式。有些复杂规则不能简单通过某一条规则指定，需要多条规则前后叠加，则使用这种匹配方式。比如参考文献

输出文件

输出文件是json结构：

- 最外层是map，包括`stat`和`detail`两个
- `stat`是用来可视化树图的数据

1. stat[].name，翻译锚点的名称
2. stat[].path，翻译锚点的名称
3. stat[].value，翻译锚点的名称
4. stat[].chidren，翻译锚点的名称

- `detail`是用来列表展示详情的数据

1. detail[].name 翻译锚点的名称
2. detail[].mode 翻译锚点识别模式
3. detail[].stat 翻译锚点统计模式
4. detail[].obj 具体的识别检测结果
5. detail[].input_ori_file 原文文件名称
6. detail[].input_trans_file 译文文件名称
7. detail[].c_origin 原文预览
8. detail[].c_trans 译文预览

下面是json文件的示例：

```
{
    "stat": [
        {
            "name": "字符和符号",
            "path": "/字符和符号",
            "value": 588,
            "children": [
                {
                    "name": "【",
                    "path": "/字符和符号/【",
                    "value": 169
                },
            ]
        },
    ],
    "detail": [
        {
            "name": "字符和符号",
            "mode": "mark",
            "stat": "poly",
            "obj": "【",
            "input_ori_file":
"data/JP/2014/JP102014000263706JP00020161241010AFULJA20160711JP005.XML",
            "input_trans_file":
"data/JP/2014/JP102014000263706JP00020161241010AFULJA20160711JP005_trans.X
ML",
            "c_origin": "<base:Paragraphs>【課題】　軸筒の前方に配",
            "c_trans": "<base:Paragraphs>已知有在配设于轴筒的前方且"
        }
    ]
```

可视化工具

根据配置文件，把对应的分类列在右侧

- 使用方式：

1. 选择检测工具输出的*_visual文件夹，如us_visual/
2. 点击“开始分析”

- 分为概览页和错误类型页：

1. 概览页做总体的饼图显示；
2. 每个错误类型页包括树图和表格两部分；

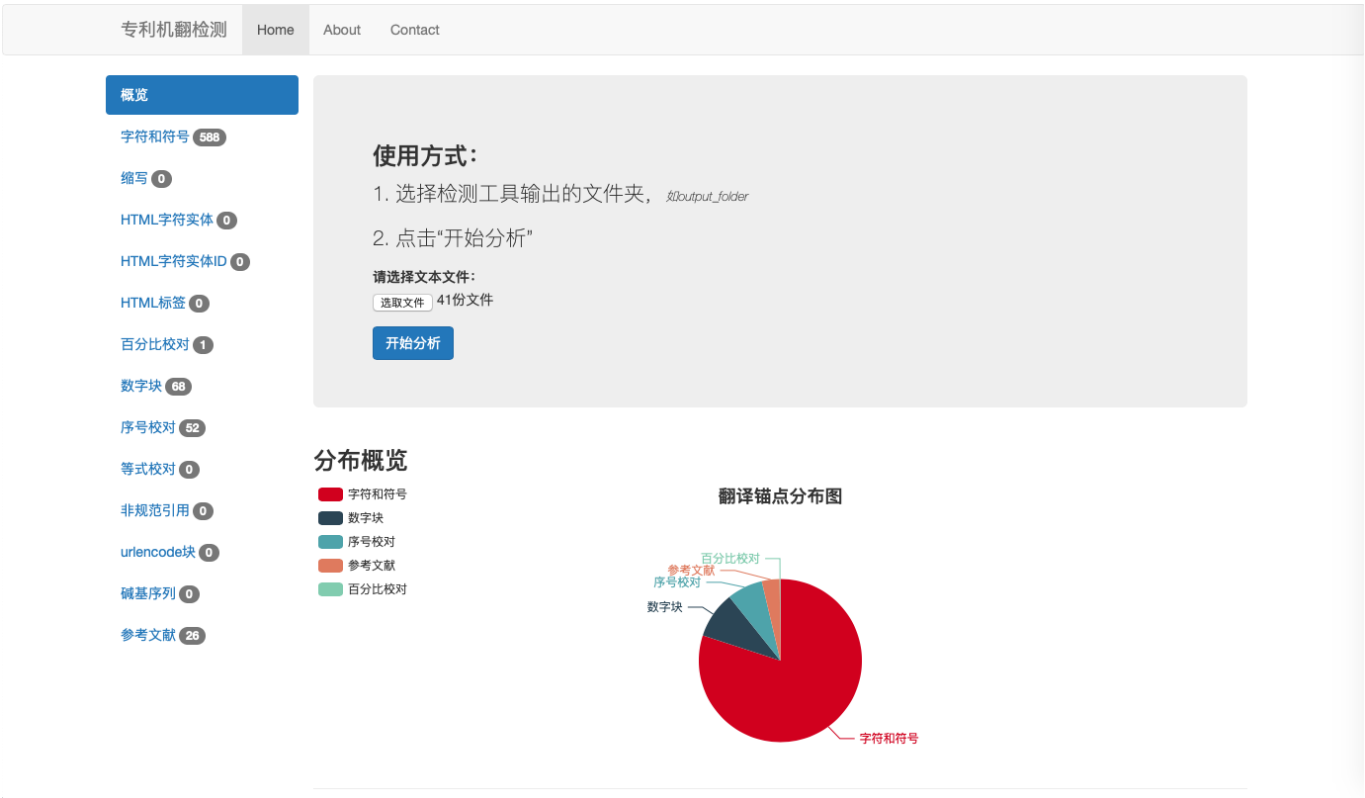
- 在树图中，用户可以直接点击树图，表格中将实时展示详细内容

- 在表格中：

1. 用户也可以直接在表格的搜索框中进行检索，查看结果
2. 简单配置一下数据目录前缀，可以点击原文和译文链接直接跳转查看原文
3. 数据目录前缀，是数据目录相对于index.html的位置，比如`../`

可视化工具是一个HTML页面，以及配套的css和js文件，目录结构如下：

```
├─index.html
├─utils.js
├─css
├─fonts
└─js
```



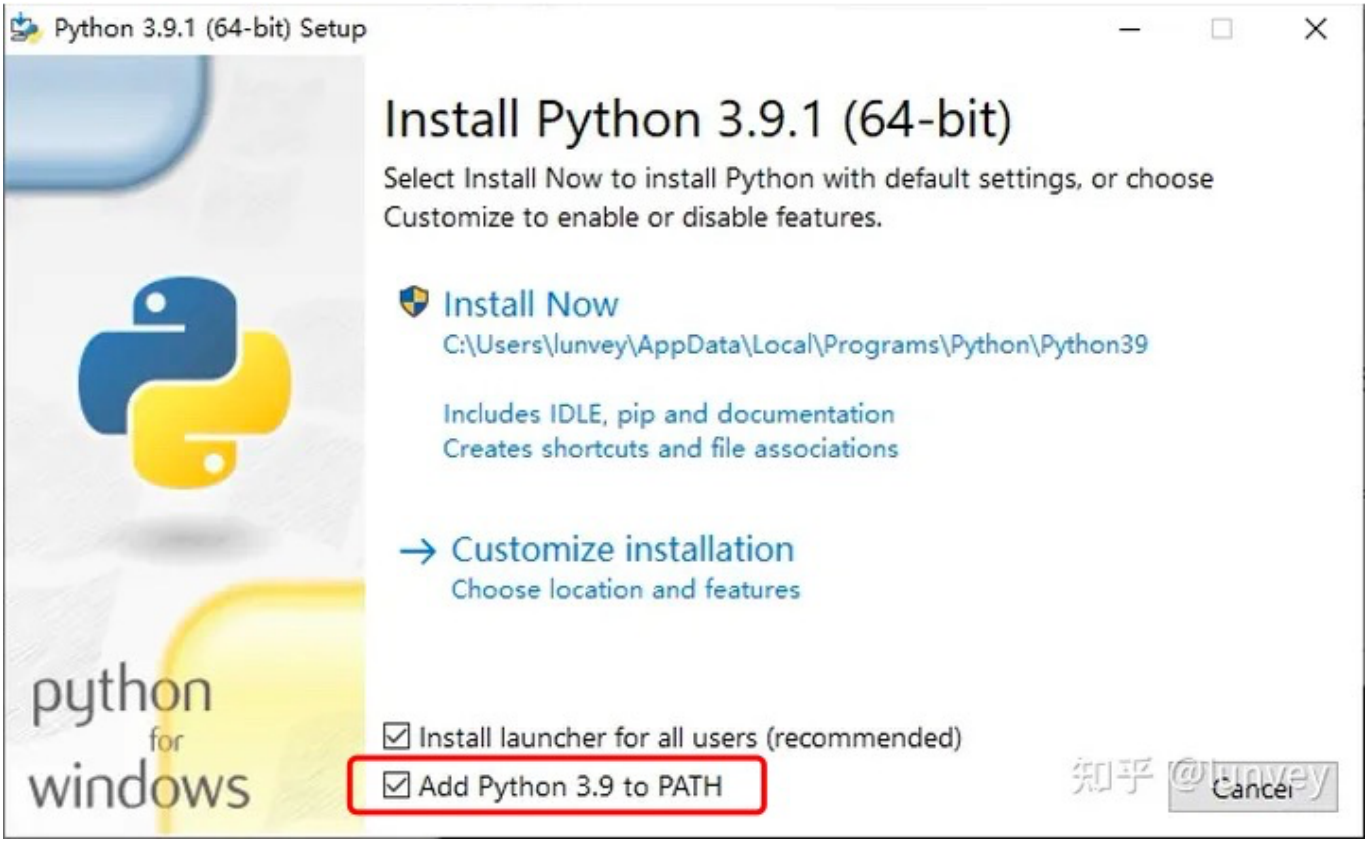
专利机翻检测工具 - 常见问题

1. 安装完python后，检测版本时仍然提示"不是内部命令"

这种情况下，可以关掉cmd再重新打开一次，之后再试一试。

如果还是不可以，需要卸载重新安装python

并且再次进入安装界面的时候，需要选择这个Add Python to PATH



2. 安装时，提示`No module named '...'`

如果缺少的模块，继续使用pip install进行安装：

```
pip install <缺少的模块名>
或
pip3 install <缺少的模块名>
```

pip是python的包管理程序，就是为了安装python包的

比如regex和lxml都是python包，用来完成一些任务，里面有功能可以调用

此外，如果安装包不成功，可以试这个pip的更新命令

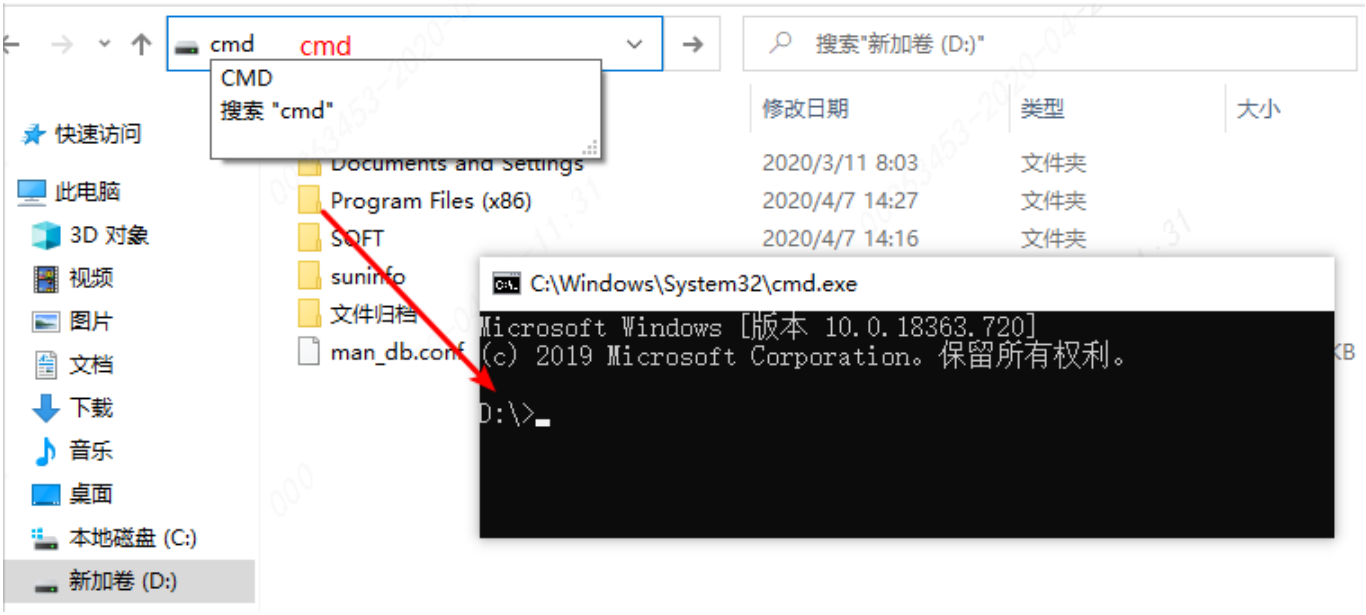
```
python -m pip install --upgrade pip
```

3. 命令`python run.py ...`应该是在哪个界面执行?

该命令的含义是，使用python解释器运行cmd当前目录的run.py文件

cmd有“当前目录”的概念，run.py需要在“当前目录”下，因此要把cmd切换到run.py所在的目录，切换方法是：

```
在资源管理器的地址栏直接写cmd
```



这样打开可以避免使用cd命令，比较方便

4. 可视化工具中表格里的搜索功能怎么用?

搜索的是字段内容，

是直接输入需要搜索的内容就可以，在下面的每一个字段内搜索。

这些表格内容里的数字和文字都可以。

5. 可视化工具中表格里的数据目录前缀该如何设置?

数据目录就是，这个index.html相对于数据的位置，使用目录结构来表示，

比如，folder_name/data和folder_name/views，那么在这里就输入`../`

`..`的含义是相对于当前目录的上级目录

输入之后，程序会把<数据目录前缀>和<input_folder/xxx.xml>拼在一起，变成

```
../input_folder/xxx.xml
```

到这个地址去找xml

6. 什么是人工智能技术？咱们的翻译锚点识别，可以称为人工智能技术吗?

人工智能，通俗来说是指让机器像人一样做决策，所以概念很广，

实现方法上，一般可以分为规则系统和统计模型两类

翻译锚点采用规则的方法来弥补统计模型的不足，不需要标注数据、可解释性强。

也就是说，翻译锚点识别可以称为人工智能技术的应用

7. 输入参数中，`-i、-o、-c、-j`分别是什么含义?

这些都是程序输入的参数名，可以按照文档说明写，也可以根据实际情况进行配置

-i，全称是--input_folder，就是输入文件夹 -o，全称是--output_folder，就是输出文件夹 -c，全称是--config，就是配置文件 -j，全称是--jobs，是指程序运行时占用的进程数，一般不需要更改。进程数越多一般来说越快，但是不能超过机器的最大核心数。