交互式数据标注工具（annot.py）使用说明

作者：牛天睿

1、基本概念

annot交互式数据标注工具是一个用于本实验室多轮RE数据标注的交互式辅助工具，它可以将一个由数字和大/小写字母组成的“*极简范式*”转译成`@李凌璇`所创的“*简化范式*”。比如，对于以下标注序列：

> girl on right with her arm out on the elephant

标注完成后的简化范式为：

> NP(* girl)PP(on right)PP(with her arm out on the elephant)

与该简化范式等效的“极简范式”仅包含四个字符：

> N2pP

annot工具的工作就是将用户输入的“极简范式”翻译成“简化范式”，从而免去人工括号匹配、数据格式化等烦冗工作，极大提高数据标注效率。

目前annot工具尚处在公开测试阶段，请在使用annot前先备份您已经标好的程序。

2、极简范式

基本的极简范式的语法为：

<头词下标>h<游标1><标签2><游标2><标签2> ...

其中<头词下标>与<游标>为阿拉伯数字，而标签为单个英文字母。每个标签都与极简范式中的一个句法标签向对应，对应表为：

```
a -> ADJP
f -> ADVP
h -> HEAD
n -> NP
p -> PP
s -> SBAR
v -> VP
```

比如，`0h0n9p`、`2h0a2n8p`等，都是合法的极简范式。

> 注意上述例子中字母h的位置，它指示了头词的位置。

接下来就为读者解释如何为一个句子标注极简范式。下面展示了一个例句，包括每个词在句子中所属的下标。

例1

girl	on	right	with	her	arm	out	on	the	elephant
0	1	2	3	4	5	6	7	8	9

留意到头词是下标0处的“girl”，因此先标注头词：`0h`。头词刚好自成一个NP。按照极简范式到简化范式的标签翻译表，我们查到NP在极简范式中用一个小写字母n替代。
而“girl”在句子中的位置为0，因此我们用`0n`把“girl”标为NP。

接着，**从左向右标注**。留意到短语“on right”属于介词短语PP，其在句子中的下标范围是1-2。由于极简范式从左向右标注，我们只需给出**短语中末尾的词的下标**，就可以指明短语在句子中的精确位置，即这里只需要给出“right”的下标（2）即可定位短语“on right”。同时，在翻译表中可查到PP对应的极简范式标签为小写字母p。因此，我们只需用`2p`即可把“on right”标为PP。

继续，后面的“with her arm out on the elephant”也构成介词短语PP。同理，我们可以用极简范式`9p`把整个短语标为介词短语。

因此，对于整个句子的极简范式标注为：`0h0n2p9p`。

对应简化范式：`NP(* girl)PP(on right)PP(with her arm out on the elephant)`

再举一个更加复杂的例子：

例2

pretty	black	girl	on	right	with	her	arm	out	on	the	elephant
0	1	2	3	4	5	6	7	8	9	10	11

在这个例子中，“girl”前面多了定语“pretty black”。因此“girl”虽然是头词，但却不存在于句子中首个句法单元中。

但按照极简范式的规则，我们仍需首先标好头词“girl”的位置：`2h`

接着，从左向右，“pretty black”是形容词短语ADJP，其对应的极简范式标签为a，因此可把它标为“1a”。

> 注意，"1"是该短语最后一个单词单词black的游标。

然后，“girl”自成名词短语NP，把它标为极简范式为：`2n`

最后，按照例1中的方法，把剩下的部分标注为两个PP：`4p11p`

最后，整个句子的极简范式标注为：`2h1a2n4p11p`

对应简化范式：`ADJP(pretty black)NP(* girl)PP(on right)PP(with her arm out on the elephant)`

进一步简化——利用首尾宏展开

上述“极简范式”的基本形式虽然相比简化范式已经有所简化，但是仍然存在诸多冗余信息。比如，例1中，“girl”既是头词，也是该句子中的第一个名词短语NP句法单元。可否用一个字符来同时表明“头词”和“NP”两个语义呢？
另外，“with her arm out on the elephant”这个短语一直延续到句子末尾，可否用单个字符来表示句子`末尾`，而免去手工输入游标“9”呢？

为了解决上述问题，我们引入极简范式中的高级语法——“首尾宏”。其规则有四条：

1. 如果一个极简范式以一个大写字母开始，则将该大写字母展开成下标为0的头词标注以及下标为0的、该字母对应的小写字母。比如，`N3p`可以展开成`0h0n3p`，`A3n5v`展开成`0h0a3n5v`。
2. 如果一个极简范式以一个游标m加一个大写字母开始，则则将该大写字母展开成下标为m的头词标注以及下标为m的、该字母对应的小写字母。比如`3N5p`展开成`3h3n5p`，而`1A3n5v`展开成`1h1a3n5v`。
3. 如果一个极简范式以一个大写字母结束，则将该大写字母展开成句子末尾的单词所在的游标以及该字母对应的小写字母。比如对于一个长度为9的句子，`0h0nP`展开成`0h0n9p`。
4. 上述规则中1与3、2与3可以同时使用。比如对于一个长为19的句子，`NP`展开成`0h0n19p`。

因此，例1中的极简范式标注可以进一步简化为`N2pP`，而例2中的极简范式标注可简化为`2h1a2n4pP`。

> 由于大多数头词都在首位或第二位，善用首尾宏可以很大程度上降低标注所需的字符数，从而极大地提高标注效率。

3、annot程序说明

3.1、环境与调用方法

annot可在Python2或Python 3环境中运行。运行方法为：

```
python annot.py your_data.txt
```

或者：

```
chmod +x annot.py # 仅运行一次
./annot.py your_data.txt
```

> 再次提醒，请预先备份您已经标好的数据！

我们推荐用`rlwrap`包裹annot程序以获得更好的编辑体验（支持上下左右导航键等）：

```
sudo apt install rlwrap
rlwrap ./annot.py your_data.txt
```

3.2、使用方法

```
➜ annot rlwrap ./annot.py COCO_牛天睿.txt
Welcome to COCO-RE Annotator!

[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE>
```

初次运行后您会看到上述界面。
第一行`[0] ID: 6794_2` 列出了当前标注的是第几个样本，以及该样本的ID。
第二行显示了当前正在标注的句子，而第三行显示了句子中每个词的下标。
您可在提示栏`CONSOLE>`后面直接输入上述极简范式，并回车。

比如：

```
➜ annot rlwrap ./annot.py COCO_牛天睿.txt
Welcome to COCO-RE Annotator!

[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> 2h1aP
Macro expand: 2h1a7p
ADJP(top right)PP(* donut on the bottom tray)
Is this ok? [y]>
```

其中，`Macro expand: 2h1a2n7p` 为首尾宏展开的结果，而 `ADJP(top right)NP(* donut)PP(on the bottom tray)` 为annot翻译得到的简化范式。

此时提示栏询问用户是否对翻译结果满意。如是，则再次按下回车，如否，则输入一个不是“y”的字符（如“n”），程序会再给您一次重新输入的机会。如图：

```
[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> 2h1a2nP
Macro expand: 2h1a2n7p
ADJP(top right)NP(* donut)PP(on the bottom tray)
Is this ok? [y]> n
Rejected last annotation, try again.
CONSOLE>
```

3.3、特殊情况

1. 不会标注或者句子本身无法标注

可以输入"!"来把当前句子标注成一个感叹号。

```
[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> !
!
Is this ok? [y]>
```

2. 不会标注或者句子本身无法标注，但我想要给出一个可能的方案

可以在输入极简范式后在尾部拼接一个"!"，来把您的答案标注为可能方案：

```
[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> 2h1a2nP!
Macro expand: 2h1a2n7p
ADJP(top right)NP(* donut)PP(on the bottom tray)!
Is this ok? [y]>
```

3. 我想要修改原句

在提示栏`CONSOLE>`后输入“rw”（意为rewrite，重写），进入重写模式，然后输入修改后的句子，最后在新的句子上标注：

```
[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> rw
REWRITE> donut on the bottom
[0] ID: 6794_2
donut on the bottom
0     1  2   3
CONSOLE> NP
Macro expand: 0h0n4p
NP(* donut)PP(on the bottom)#!
Is this ok? [y]>
```

4. 极特殊，如双头词的情况

在提示栏`CONSOLE>`后输入“m”（意为manual，手动），进入手动模式，手动输入句子标注：

```
[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> m
MANUAL> NP(* donut)
NP(* donut)
Is this ok? [y]>
```

3.4 保存数据与其他功能

在提示栏`CONSOLE>`后可输入如下几个annot所提供的合法命令：

- s（或save）：保存当前的标注成果到文件中。程序会提示您输入一个文件名，默认为覆盖原始文件。
- h（或help）：列出所有内置命令与标签。
- Q（或quit）：不保存，退出。
- auto（或autosave）：打开/关闭自动保存。自动保存打开的情况下，用户每标注一个数据，就会把整个数据库写入**默认文件**。默认文件可以通过save命令来修改。此时，提示栏会有[AUTO]字样：

```
[0] ID: 6794_2
top right donut on the bottom tray
0   1     2     3  4   5      6
CONSOLE> auto
Autosave: On.
The annotations will be saved at: annotation.txt
CONSOLE[AUTO]>
```

- p（即print）：重新打印当前在标注的样本。
- u（即up）：回到上一个标注的样本。**注意：这会删除上一个样本的标注结果！**
- d（即down）：进到下一个要标注的样本。**注意：这会删除下一个样本的标注结果！**

3.5 重新载入上次的标注结果

在您第二次启动时，annot程序会自动跳过已经标注过的样本，返回上次标注的上下文，

以便您继续标注：

```
Skipped the [45]-th sample.

Skipped the [46]-th sample.

Skipped the [47]-th sample.

Skipped the [48]-th sample.

Skipped the [49]-th sample.

Skipped the [50]-th sample.

[51] ID: 7058_0
girl on right with her arm out on the elephant
0    1  2     3    4   5   6   7  8   9
CONSOLE>
```