自然语言提取流程解析及第一次测试结果

安昊哲

2017.8.25

一. 简介

自然语言处理，英文为 Natural Language Processing，简称 NLP，是将人类日常交互语言转变为计算机容易理解的、机构化的格式，从而可以更加快速和准确地处理各种信息。

这项功能将会应用于关于语音输入的项目中，例如导诊机器人分诊的功能就需要分析患者对其疾病与症状的描述，转化为相应的格式，再输入到后台程序。

二. 应用技术

哈工大 pyltp 分词包：其官网介绍，“pyltp 是 LTP 的 Python 封装，提供了分词，词性标注，命名实体识别，依存句法分析，语义角色标注的功能”。在 Extractor.py 中所应用的功能主要为分词和依存句法分析，二者相结合以提取出相关的医疗信息。

关于 pyltp 的详细讲解，请阅读官方网站的文件：
http://pyltp.readthedocs.io/zh_CN/latest/api.html#pyltp

三. 自然语言提取的内容及其输出格式

a. 在一句输入中，我们所感兴趣的、关键的医疗信息包括出现异常的人体器官（如头部、心脏、胃、腿等），人体组织（如血液、皮肤、淋巴等），人体其他指标（如尿液、大便、视力、睡眠等），病灶的方位（在器官的上、下、左、右等），主要的问题和人体感觉（如疼、痒、麻、肿胀、骨折等），其问题形式，问题的严重程度，问题出现的缓急、频率和时长，问题起因，加剧问题的因素，缓解问题的因素，病人的病史，其怀疑疾病，排除疾病以及其他否定描述的症状。

b. 我们将以上信息提取，根据问题的个数，输出一个或多个结构化的 Tuple，其格式为：
（[器官 A，器官 B ...]， [[器官 A 方位],[器官 B 方位] ...]， [人体组织], [人体指标]， *[问题或感觉]*， *[问题或感觉的形式]*,
int(问题或感觉的程度)， [不包含症状]， int(症状出现缓急),
int(症状出现频率)， int(症状持续时间),
str(起因，暂不填写)， str(加剧因素，暂不填写)， str(缓解因素，暂不填写),
[病史]， [怀疑疾病]， [排除疾病] ）

请注意，每一个症状的问题都会有一个对应的 Tuple。因此只有上方格式中斜体部分是每个 Tuple 独特的结果，其余部分为每一句主诉所有输出 Tuple 的共享内容。

c. Tuple 的每一部分都对应着一个关键词词典和一个英文缩写，其对应关系如下：

Index in Tuple	Element	Dictionary Name in Github	Acronym
0	器官	organ_nlp	O
1	方位	location_nlp	L
2	人体组织	tissue_nlp	T
3	人体指标	indicator_nlp	I
4	问题或感觉	problem_nlp + feeling_nlp	P + F
5	问题或感觉的形式	暂无	暂无
6	问题或感觉的程度	severity_nlp	N/A
7	不包含症状	symp_nlp + negative_nlp	N/A
8	症状出现缓急	suddenness_nlp	N/A
9	症状出现频率	frequency_nlp	N/A
10	症状出现时间	time_nlp	N/A
11	症状持续时间	time_nlp	N/A
12	起因	暂无	暂无
13	加剧因素	暂无	暂无
14	缓解因素	暂无	暂无
15	病史	disease_nlp + history_nlp	D
16	怀疑疾病	disease_nlp + suspect_nlp	D
17	排除疾病	disease_nlp + eliminate_nlp	D

Table 1: Tuple 结构中各成分与其词典和缩写的对应关系及简称

目前，词典内容已包含大量的词汇，但并不是完全没有疏漏，需要在日后的测试中加以完善。

d. 由于技术尚未成熟，当前的程序还不能提取自然语言中关于疾病起因、加剧因素和缓解因素的成分。这些内容将根据其他产品的需求，酌情考虑是否应添加至 Tuple 的格式中。

e. Tuple 中的一些成分是转化为整数(int)作为输出的，其对应关系如下：

症状出现频率的等级划分	
总是	3
经常	2
有时	1
未出现	-1
症状程度的等级划分	
严重	3
适中	2
轻微	1
未出现	-1
症状出现缓急的等级划分	
急	2
缓	1
未出现	-1
出现时间及持续时间等级划分	
time > 三个月	7
一个月 < time <= 三个月	6
一星期 < time <= 一个月	5
三天(72h) < time <= 一星期	4
一天(24h) < time <= 三天(72h)	3
2h < time <= 24h	2
time <= 2h	1
未出现	-1

Table 2: Tuple 中整数输出的等级划分

举例说，当我们输入主诉“睾丸不舒服，下坠。小腹也不舒服。腰部也会不舒服”时，程序应输出结果：

(['O_睾丸', 'O_小腹', 'O_腰部'], [[ ], [ ], [ ]], [ ], [ ], ['P_不舒服'], [ ], -1, [ ], -1, -1, -1, '起因暂无', '加剧因素暂无', '缓解因素暂无', [ ], [ ], [ ])

(['O_睾丸', 'O_小腹', 'O_腰部'], [[ ], [ ], [ ]], [ ], [ ], ['P_下坠'], [ ], -1, [ ], -1, -1, -1, '起因暂无', '加剧因素暂无', '缓解因素暂无', [ ], [ ], [ ])

在以上输出结果中，首字母和 Table 1 中的 Acronym（简称）所对应。并且，就像之前解释过的，只有病状问题和形式是每个 Tuple 中独特的元素，其它成分均为共同内容。

四. 自然语言提取流程

所有自然语言提取的程序代码均在 Extractor.py 中。

a. 人体器官、组织和其他指标的提取
   i. 利用 ltp 将自然语言分词
   ii. 在分词结果中，一一查询是否在其对应的词典中，若存在，即提取出作为结果，并标注相对应的英文简称
b. 器官方位的提取
   i. 利用 ltp 将自然语言分词
   ii. 在找到器官词汇时，寻找与其构成“定中关系”(ATT，详见 pyltp 官方文档中关于依存句法的解释)的词汇，存储这些词汇，称之为 X
   iii. 寻找和 X 构成“并列关系”（COO）的词汇，存储这些词汇，称之为 Y
   iv. 在 X 和 Y 中选取对应方位词典的词汇，加入 Tuple。若该词汇不在词典中，暂时也加入了 Tuple 中，可以用来检测词典的完整程度，弥补漏洞。
c. 症状问题和感觉的提取
   i. 利用 ltp 将自然语言分词
   ii. 寻找句子的根节点和与其构成“并列关系”（COO）的并且含义不同的词汇分别添加至新创建的 Tuple 中，这些词汇有很大几率是病灶的描述词语，但不排除有噪音的可能
   iii. 在第二步完成后，一一查询分词结果中的每个词汇，若其存在于相对应的词典并且未在第二步中抓取到，则分别添加至新创建 Tuple 中
d. 问题形式的提取
   i. 根据问题提取的每一个结果，找寻与其构成“状中结构”(ADV)和“动宾关系”(VOB)的词汇，检查其是否在方位词典中，若不在，添加至 Tuple 对应的 list 中，称之为 X
   ii. 检查和 X 构成“并列关系”（COO）的词汇，重复第一步关于方位的检查，将第一步没有添加的词汇加入到 Tuple 中
e. 问题程度、缓急、频率和持续时间的提取
   i. 利用 ltp 将自然语言分词
   ii. 在分词结果中，一一查询是否在其对应的词典中，若存在，及提取出词典中相对应的数值，加入到 Tuple 中
f. 不包含问题的提取
   i. 利用 ltp 将自然语言分词
   ii. 调动 negative_nlp 的词典，找出包含否定含义的词汇
   iii. 若第二步中找到了否定词汇，则利用一个 recursive 的方法，抓取所有否定词汇的子节点，将其加入 Tuple 中并标注相对应的英文简称
   iv. 值得注意的是，此方法提取出大量噪音，需要利用更完善的词典来进行筛选

g. 病史、怀疑疾病和排除疾病的提取
    i. 利用 ltp 将自然语言分词
    ii. 调动相对应的词典，找出句子中表达意思的关键词
    iii. 在每一句包含关键词的句子中，一一查询是否存在词典中的疾病名称，若存在，添加至相对应的 list，并标注英文简称

五. 测试评分方法

每条主诉总计 8 项测试内容，共 100 分。
第一项：人体器官、组织和其它指标，总分 30。

- 假设这三个集合中共有 x 个输入，则每项分值为 35/x。若提取结果中包含对应项，则得 35/x 分。
- 对比正确结果与提取结果的元素个数，提取结果中多出的，每个扣 5 分。

第二项：方位，总分 5。

- 假设这三个集合中共有 x 个输入，则每项分值为 5/x。若提取结果中包含对应项，则得 5/x 分。
- 对比正确结果与提取结果的元素个数，提取结果中多出的，每个扣 1 分。

第三项：问题与感觉及其形式，总分 35。

- 问题与感觉部分分值为 30。
- 假设这个集合中共有 x 个输入，则每项分值为 30/x。若提取结果中包含对应项，则得 30/x 分。
- 对比正确结果与提取结果的元素个数，提取结果中多出的，每个扣 3 分。

- 形式部分分值为 5。
- 若在此集合内发现了正确结果中的“问题与感觉”的某项，则增加 0.6 * 30/x 分。
- 假设这个集合中共有 x 个输入，则每项分值为 5/x。若提取结果中包含对应项，则得 5/x 分。
- 对比正确结果与提取结果的元素个数，提取结果中多出的，每个扣 1 分。

- 在本项中，若提取结果和正确结果相差甚远，可能出现分值为负数的情况。

第四项：程度、缓急、频率和时间，总分 20。

- 每项 5 分，提取数值一致即得分，否则不得分。

第五项：不包含症状，总分 4。

- 假设这个集合中共有 x 个输入，则每项分值为 4/x。若提取结果中包含对应项，则得 4/x 分。

第六项至第八项：病史、怀疑疾病和排除疾病，总分 6。

- 每项 2 分，提取结果一致即得分，否则不得分。

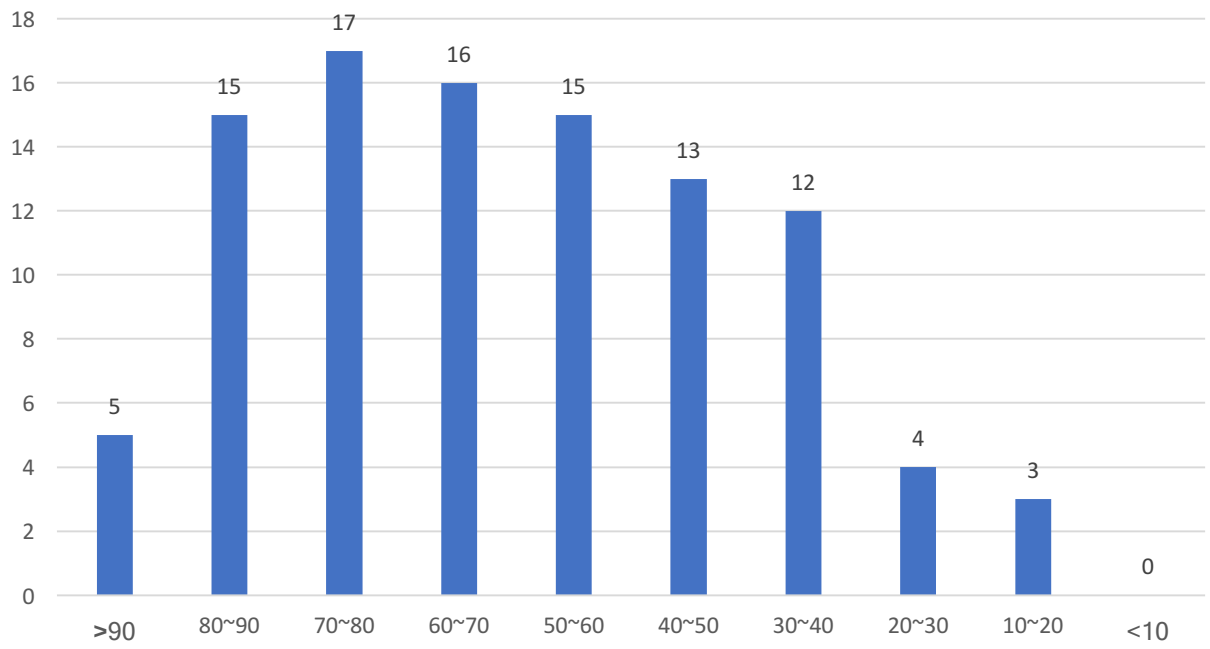
六. 第一次测试结果

所有测试代码均在 test.py 中

a. 测试数据来源于最初在好大夫网站上收集的患者主诉，从中随机抽取了 100 条后，人工标注出最理想的提取结果，存在“input_nlp_test”文本中。其中，‘N’代表空集，内容顺序与 Tuple 的成分一致。

b. 根据上文的评分方法，100 条测试的平均分为 60.29，中位数为 61.50，其具体分布图如下

Figure 1: 第一次测试结果得分分布图



七. 问题与挑战

虽然已经实现了将自然语言转化为结构化的 Tuple 的形式，但在提取过程中仍然存在一些问题。

a. 同义词的理解尚未添加至当前的程序中，对于“肚子疼”和“腹痛”这样同样意思的表达方式，会理解为完全不同的两个 Tuple。这个问题的可以通过之前用

Aho Corasick 的方法进行近义词替换，需要用到之前所写的 Match.py 的功能。尚未尝试添加此方法，猜测有可能会因为 Match.py 中对近义词替换的方法不够有效率而导致代码运行速度大大减缓。

b. 对于提取结果噪音的处理取决于我们对现存词典完整情况的自信程度，当词典不够完整时，目前“宁多勿漏”的提取方法可能有助于我们发现漏洞，弥补词典中的不足。当词典足够完整时，我们可以利用词典作为筛选的标准，进而增加提取结果的准确性。

c. 评测集中可能出现少量标注错误，例如忽略了疾病程度的词汇等情况，进而导致测试结果中小部分得分偏低。解决此问题的方法为人工对比标注结果和评测结果，更改标注中的错误。

d. 测试评分方法为本人一人评定，因此主观性较大，可能无法代表全部测试的准确性。