知识获取任务说明

1. **任务说明**

给定一个文档，从中抽取可能存在的三元组，三元组以（实体1，关系，实体2）或者（实体，属性，属性值）的形式存在。文档在语言方面不做限制，方法可以是基于依存分析的，也可以是基于神经网络的。

1. **开发环境**

**开发语言：**Python 3.6；

**开发环境：**PyCharm；

**第三方库：**pyltp。pyltp是LTP的python封装，提供了分词、词性标注、命名实体识别、依存分析等功能，安装过程如下：

（1）安装Visual C++ 2015 Build Tools；

（2）执行pip install pyltp。如果不成功则下载相应的whl文件，下载网址：[https://www.mlln.cn/2018/01/31/pyltp%E5%9C%A8windows%E4%B8%8B%E7%9A%84%E7%BC%96%E8%AF%91%E5%AE%89%E8%A3%85/pyltp-0.2.1-cp36-cp36m-win\_amd64.whl，然后在其目录下执行pip install pyltp-0.2.1-cp36-cp36m-win\_amd64.whl](https://www.mlln.cn/2018/01/31/pyltp%E5%9C%A8windows%E4%B8%8B%E7%9A%84%E7%BC%96%E8%AF%91%E5%AE%89%E8%A3%85/pyltp-0.2.1-cp36-cp36m-win_amd64.whl，然后在其目录下执行pip%20install%20pyltp-0.2.1-cp36-cp36m-win_amd64.whl)；

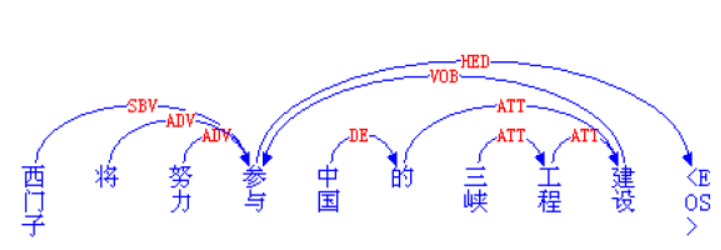
（3）下载pyltp所需要的模型，下载网址：<http://ltp.ai/download.html>。

1. **方法说明**

本实验可基于依存分析或者神经网络实现，基于依存分析的方法对于文档限制不高，不需要训练数据，能够抽取各种类型的关系，但是容易引入噪音，而基于神经网络的方法需要高质量的训练数据，对于关系类型也有限制，但是如果模型训练得当，将会有一个比较高的精确度和召回率。实验可任选两种方法之一完成：

**3.1 基于依存分析**

依存分析是NLP中的关键技术之一，其基本任务是分析并确定句子中词汇之间的相互依存关系。比如句子“我送她一束花”，“我”与“送”构成主谓关系（SBV），“送”与“花”构成动宾关系（VOB），“送”与“她”构成间宾关系（IOB）；再比如，“红苹果”中的“红”与“苹果”构成定中关系（ATT），“非常美丽”中的“非常”与“美丽”构成状中关系（ADV）等等。依存分析的结果往往以一个树的形式展现：



经过对句子的依存分析，我们可以基于依存分析的结果对句子进行三元组的抽取。比如“小明喜欢跑步”，可以根据主谓关系（SBV）和动宾关系（VOB）的关系提取出三元组（小明，喜欢，跑步），“张三和李四的朋友”可以根据主谓关系（SBV）和并列关系（COO）提取出三元组（张三，朋友，李四）。同学们可以自行挖掘其他可以提取出三元组的依存关系组合。

由于依存关系分析是对句子进行分析，而文本是非结构化的数据，因此提取出来的三元组可能并不是真正的三元组，比如其头部元素可能并不是一个实体，因此需要引入额外的词典用以判断某个三元组是否是对于某个实体的描述。比如“我喜欢跑步”，这个句子和“小明喜欢跑步”的区别在于，“我”并不是一个合法的实体指代，而词典中也不会包含“我”这个词，这样头部元素是“我”的三元组就会被过滤掉，进而提高关系抽取的准确率。

除了通过引用外部字典的方式，我们也可以通过命名实体识别的方式，对产生的三元组进行约束。即先识别出句子中的实体，然后通过依存分析，进行相应的三元组抽取。

依存分析可用现成工具实现，比如Stanza、Hanlp、PyLTP等等。示例代码中提供了一个基于依存分析的三元组抽取demo，使用的依存分析工具是PyLTP。

**3.2 基于神经网络**

除了基于依存分析的方法，本实验也鼓励同学实现基于神经网络的关系抽取任务。由于现在NLP领域对于基于神经网络的关系抽取方法已经有了很多研究进展，模型五花八门，因此本实验不提供基于深度学习的示例程序。如果有感兴趣的同学想做基于该方法的关系抽取，则本实验不对数据、抽取的关系类型、文档类型、文档语言种类及神经网络模型做限制，同学可自由发挥。

1. **示例程序文件说明**

**pyltp\_models：**该文件夹包含了pyltp库所需要的几个模型。由于模型过大，为了方便从github下载示例程序，该文件夹已被清空。请同学们在运行示例程序前先在工程目录中创建该文件夹，然后到<http://ltp.ai/download.html上下载3.4.0>版本的模型到该文件夹中。

**construct\_dict.txt：**字典文件，用来对三元组进行约束，筛掉一些不包含实体的三元组。

**input.txt：**输入文件，将需要提取三元组的文档复制到该文件中。

**output.txt：**输出文件，被提取的三元组会被输出到该文件中。

**sentences.txt：**中间处理文件，用以存储输入文档的分句结果。

**knowledge\_extraction.py：**python文件，程序主体。

**data.nt:** 输出文件，output.txt文件经过处理转化成data.nt文件，形式上更为标准