维基百科人物属性自动获取方法研究*

孟新萍[1,2] 王会珍[1,2] 张俐[1,2]

1. 东北大学自然语言处理实验室，辽宁沈阳，110004

2. 医学影像计算教育部重点实验室（东北大学），辽宁沈阳，110819

E-mail: mengxp@ics.neu.edu.cn

摘　要： 人物属性抽取是人名搜索引擎和社会关系网络构建的重要基础。本文提出了一种从维基百科中自动获取人物属性的方法。该方法利用人物类维基文本现有信息盒中的“人物姓名-属性-值”三元组关系，对给定的每个属性，将人物姓名和属性值标记到维基自由文本中，自动生成带标注的数据集。利用该数据集使用机器学习的方法自动生成模板，通过模板匹配从维基文本中获取更多的属性信息，同时也达到了生成完整的信息盒的目的。实验证明，该方法可以有效的抽取出人物的属性。

关键字： 维基百科；人物属性抽取；模板自动获取

Study on Automatic Person Attribute Extraction from Wikipedia

Meng Xinping[1,2], Wang Huizhen[1,2], Zhang Li[1,2]

1. Natural Language Processing Lab, Northeastern University, Shenyang, 110004

2. Key Laboratory of Medical Image Computing (Northeastern University), Ministry of Education, Shenyang, 110819

E-mail: mengxp@ics.neu.edu.cn

Abstract: Person attribute extraction is one of fundamental techniques of person name search engine and social network construction. This paper proposes a method to automatically generate the infobox of person attributes, which uses the "name-attribute-value" triples in the existing infoboxes of Wikipedia. For a given attribute, our system marks the person name and the attribute value in the corresponding sentences in free texts of Wikipedia, and automatically acquires marked data set. Patterns of each attribute can be generated automatically by machine learning algorithms. Then more attributes can be acquired by means of pattern matching, and at the same time the attributes can be used to generate complete infoboxes. Experiments showed that our method could extract person attributes effectively.

Keywords: Wikipedia; Person Attribute Extraction; Pattern Automatic Acquisition

1 引言

利用搜索引擎检索人物信息是互联网垂直搜索技术的一个典型应用。然而现实世界中多个人共享同一个名字的现象普遍存在。当用户查询所关注的某个人物的详细信息时，通用搜索引擎返回的结果往往是共享这一名字的不同人物相关网页的混合，需要用户一一判别。正是在这种背景下，人名搜索引擎技术越来越受人们关注。此外，网络上的人物信息中蕴含着大量关于人物之间关系的信息。收集特定领域内的人物关系信息构成社会网络，人们便可以沿着关系链以最快的速度找到自己感兴趣的人。人物搜索引擎和社会关系网络构建中一个重要的基础技术是人物基本属性的抽取。因此，从web中抽取人物属性成为一个重要的研究课题。

叶正[1]等把人物属性抽取作为实体关系抽取的一种具体应用，使用Hownet提取描述人物属性

* 本课题研究工作部分得到了国家自然科学基金项目（60873091）、中央高校基本科研业务费专项资金资助。

的词作为触发词，将触发词和人名间的描述关系转化为分类问题。该方法在训练分类器时需要人工标注数据，并且用到了语义资源。此外，该文中只用到了人工标注的700条数据，数据量相对较少。王颖[2]在提取人物属性时，对“个人介绍类”文本使用了知识工程的方法进行人物属性的抽取。在观察总结大量网页内容和研究自然语言理解的基础上，人工归纳了一些规则，建立模式规则知识库进行模式匹配。用规则的方式抽取属性，准确率很高。但规则的书写是一项很繁杂的工作，不同人写出的规则往往很难统一。Cesar de Pablo-Sanchez等[3]应用命名实体识别技术和分类技术从过滤好的web网页中选择候选属性，并用模板去匹配正确的属性值实体。模板的获取采用了三种策略，一是基于正例和负例的模板获取方法，二是基于正例的模板获取方法，三是在一二的基础上进行人工校正。在对模板进行了人工校正后，系统取得了很好的性能。

本文提出了一种从维基百科中自动获取人物属性的方法。人物类维基文本中的信息盒以表格的形式对人物的重要属性进行了描述，这给提取人物属性带来了很大方便。通过对维基百科的研究发现，信息盒中的属性多数在文本中都有对应的句子。本文提出的方法正是利用了维基百科的这一特性。该方法借助人物类维基文本已有信息盒中的 “人物姓名-属性-值”的三元组关系，将人物姓名和属性值标记到维基自由文本中，自动生成带标注的数据集。利用标注好的数据集通过机器学习自动生成获取每个属性的模板。用生成的模板到维基文本中匹配，则可以抽取出更多的属性，相当于生成或完善了信息盒中的属性信息。该方法中数据集的标注和模板的生成都是自动的，不需要手工标注数据集和人工书写模板，并且没有用到任何的语义资源。

2 任务定义

本文的研究任务是，给出人物类维基自由文本和维基中现有的人物类信息盒集合，从人物类维基文本的信息盒中选定几个属性，在自由文本中自动生成包含选定属性的人物信息盒。例如，选定 date of birth（出生时间），date of death（逝世时间），place of birth（出生地点），place of death（逝世地点），native place（籍贯）和 spouse（配偶）六个属性，并给定人物类维基自由文本和现有信息盒的集合，无论原来的维基文本中信息盒存在与否，系统都可以生成包含上述6个属性的信息盒（假定每个属性都可以在维基中找到对应的句子），当用模板没有匹配到给定的属性时，属性值的位置为空。对于“杨开慧”这一条目，系统生成的信息盒格式如图1所示：

```
<title>
杨开慧
</title>
<attribute>
<杨开慧,date of birth,1901年11月6日>
<杨开慧,date of death,1938年11月14日>
<杨开慧,place of birth,湖南省长沙县板栗乡>
<杨开慧,place of death,>
<杨开慧,natice place,湖南省长沙>
<杨开慧,spouse,毛泽东>
</attribute>
```

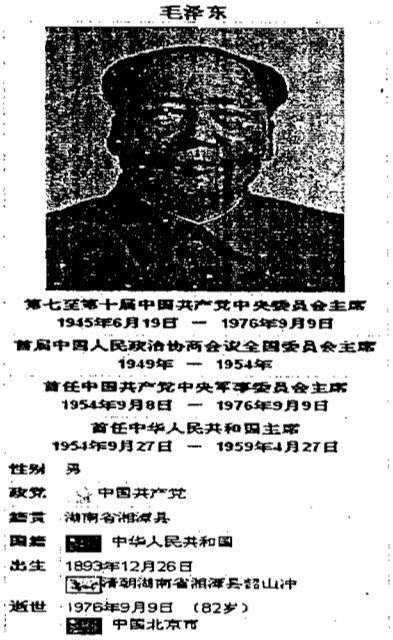
图1：系统生成的信息盒示例

从自由文本中自动抽取人物属性的关键在于如何自动构造描述人物特定属性的模板。例如定义描述人物出生日期的模板。维基文本中没有直接标注人物属性信息，但部分文本提供了信息盒的结构化信息标注。本文工作的基本思想就是利用部分信息盒来自动构造人物属性抽取模板，然后选择置信度高的模板从新自由文本中自动抽取人物属性信息，自动构造对应的结构化信息盒。

3 维基百科人物属性获取方法

3.1 从信息盒中提取三元组

维基文本中的每个信息盒都对应一个产生该信息盒的模板。维基文本中显示的信息盒和产生信息盒的模板格式如图 2 所示。从人物类维基文本的信息盒中，可以抽取出人物的一些重要属性信息，将其表示成<人物姓名，属性，属性值>的三元组形式。例如，对“毛泽东”这一条目的 Native place 属性，抽出的三元组为<毛泽东，Native place,湖南省湘潭县>。




```
{{Infobox_Politician_Basic
| name = 毛泽东
| image name = Mao Zedong portrait.jpg
| width = 228px
| now = 第七至第十届[[中国共产党中央委员会主席]]
| period = [[1945年]][[6月19日]] - [[1976年]][[9月9日]]
| now2 = 首届[[中国人民政治协商会议全国委员会主席]]
| period2 = [[1949年]] - [[1954年]]
| now3 = 首任[[中国共产党中央军事委员会主席]]
| period3 = [[1954年]][[9月8日]] - [[1976年]][[9月9日]]
| now4 = 首任[[中华人民共和国主席]]
| period4 = [[1954年]][[9月27日]] - [[1959年]][[4月27日]]
| Sex = [[男]]
| different name =
| party = {{CPC}}
| Native place = [[湖南省]][[湘潭县]]
| nationality = {{PRC}}
| date of birth = {{birth date|1893|12|26}}
| place of birth = {{flagicon|Qing Dynasty}}[[清朝]][[湖南省]][[湘潭县]][[韶山冲]]
| date of death = {{death date and age|1976|9|9|1893|12|26}}
| place of death = {{CHN}}[[北京市]]
.....
}}
```

图 2：维基百科文本中显示的信息盒和其对应的信息盒模板示例

从信息盒中提取出三元组后，需要做以下两步处理：

1) 信息盒中的属性值存在模板嵌套的形式。例如，{{birth date|1893|12|26}} 代表 1893 年 12 月 26 日，{{CHN}}代表中国。根据维基百科中模板的定义，将其转换为对应的文本形式。
2) 人物属性可能有多个并列的属性值，这里将其当作多个属性-属性值来处理。例如：“毛泽东”这一条目信息盒中的 spouse 属性含有四个属性值，spouse = #[[罗一秀]] ([[1907 年]] - [[1910 年]])#[[杨开慧]] ([[1920 年]] - [[1930 年]])#[[贺子珍]] ([[1928 年]] - [[1976 年]])#[[江青|江青]] ([[1938 年]] - [[1976 年]])，处理后表示为：
   <毛泽东，spouse1，[[罗一秀]] ([[1907 年]] - [[1910 年]])>
   <毛泽东，spouse2，[[杨开慧]] ([[1920 年]] - [[1930 年]])>
   <毛泽东，spouse3，[[贺子珍]] ([[1928 年]] - [[1976 年]])>
   <毛泽东，spouse4，[[江青|江　青]] ([[1938 年]] - [[1976 年]])>

3.2 数据集自动标注

三元组中的属性值，在维基文本中有时并不是作为一个整体同时出现的。因此，在进行属性值匹配时，将属性值划分为若干词的片段。划分片段时，要综合考虑分词和信息盒中属性值原有的标记。例如，将“毛泽东”这一条目中的 place of birth 的属性值划分为清朝、湖南省、湘潭县、韶山冲四个片段。对于选定的每个人物属性，将其对应的人物姓名和属性值片段标记到训练文本对应的句子中，抽取出同时包括人物姓名和属性值片段的句子。当属性值包含多个属性值片段的

时候，可能匹配到多个句子，根据匹配的属性值片段数进行打分，选择分数最高的句子。

3.3 模板自动获取

3.3.1 生成候选模板集

候选模板集的生成主要依赖于人物姓名和属性值片段的标记以及生成模板时选取的特征。在将人物姓名和属性值片段标记到句子中生成带标注的数据集时，根据匹配的片段数对句子打分，选择得分最高的句子。生成模板时，主要选取属性值前后的词作为特征。候选模板自动构建的基本思想是，利用现有信息盒中已知的人物属性信息从对应文本中找到包含该信息的句子，然后从被选择的句子中自动构建该人物属性的描述模板。

算法 1：生成候选模板集P_c

输入：属性类型 attr，无标注的训练文本 C 和三元组集 T
输出：候选模板集P_c
1) 从T中抽取出属性attr所在的三元组集T_a；
2) 分别用人物姓名和 attr 的属性值到 C 中去标记；
3) 抽取出人物姓名和属性值片段同时标记到的句子集 S；
4) 对S中的句子按照匹配的属性值片段数进行打分，从每一篇文本中选择得分最高的句子构成句子集S_a；
5) 分别用<name>和<attr_value>标签去替换S_a中具体的人物姓名和属性值。
6) 在<attr_value>的前面和后面分别取 m 和 n 个词，生成模板；
7) 保留模板特征词中的动词、介词、名词、数量词、助词“的”和标点，其余词泛化为任意词；
8) 若<attr_value>后面的特征词出现标点，标点之后的特征词不再保留；
9) 对只有介词、连词或标点不同的模板进行合并，生成候选模板集P_c。

3.3.2 模板的评价及最终模板集的生成

由于算法 1 生成的候选模板集P_c可能存在两个问题：噪音模板和模板冲突问题。为了解决这些问题，需要从中选择出一部分质量好的模板生成最终的模板集，排除噪音模板。这里给出模板的一个置信度评价指标，即用候选模板集P_c中每个模板在训练文本抽取出的正确的属性值的比例来表示每个模板的置信度。对于模板冲突问题的求解可以基于对抽取的答案进行评价来解决，或者通过控制模板的匹配顺序（高置信度模板优先匹配原则）来解决。

算法 2：计算候选模板集P_c中每个模板的置信度，并生成最终的模板集P

输入：候选模板集P_c，无标注的训练文本C
输出：最终的模板集 P
1) 从训练文本 C 中获取包含文本标题（人物姓名）的句子集 S；
2) 对于候选模板集P_c，分别按以下两种方式计算每个候选模板在 1）中得到的句子集S中出现的次数：
   ① <attr_value>被正确的属性值匹配的句子数 Ca；
   ② <attr_value>被任意词匹配的句子数C_o；
3) 计算每个模板的置信度P_a，其中$P_a=C_a/C_o$；
4) 获取置信度满足一定阈值 τ 的模板集 P 。

选定 date of birth, date of death, place of birth, place of death, native place, spouse 六个属性，使用算法 1 生成候选模板集，并按照算法 2 对模板进行置信度评价，按置信度高低排序后排在前 3

位的模板及其置信度如表 1 所示。算法 1 中，取 m=3，n=3，算法 2 中取 τ=70%。模板中的{.*}代表任意个数的任意词语或标点，{标点}代表任意标点，{A|B}表示 A 和 B 任选其一。

表 1：模板示例及其置信度

属性	模板	模板置信度
date of birth	<date of birth>，<name>出生{在|于}	100%
	<name><date of birth>生于	100%
	<name>(<date of birth>—	87.5%
date of death	<date of death>,<name>在{.*}去世	100%
	<name>于<date of death>在{.*}逝世	100%
	<name>({.*}—<date of death>)	88.1%
place of birth	<name>出生{在|于}<place of birth>的一个	100%
	<name>{.*}出生于<place of birth>{标点}	98.3%
	<name>{.*}生于<place of birth>{标点}	89.6%
place of death	<name>{.*}，逝世于<place of death>{标点}	98.4%
	<name>{在|于}<place of death>去世{标点}	97.3%
	<name>因病于<place of death>逝世{标点}	89.2%
native place	<name>{.*}原籍<native place>{标点}	100%
	<name>{.*}祖籍<native place>{标点}	100%
	<name>{.*}，<native place>人{标点}	86.4%
spouse	<name>{与|和}<spouse>育有{.*}	100%
	<name>{.*}{与|和}<spouse>{在|于}{.*}结婚	100%
	<name>{.*}是<spouse>的{.*}妻子	100%

3.4 基于模板的人物属性获取

对每个特定的属性，用模板集中的每个模板分别到自由文本中去匹配。匹配时，优先选择置信度高的模板。当某一属性能通过置信度高的模板匹配抽取出来时，则不再使用其他置信度低的模板到该条目对应的文本中去匹配。这在一定程度上降低了模板匹配到的无关句子的数目。

用模板匹配时，首先匹配人物姓名（对应文本的标题），然后匹配模板中属性两端的特征词，对于{.*}的匹配可以采用正则表达式匹配的方法。

4 实验与结果分析

4.1 数据集

本文实验中的数据集采用维基百科 2010 年 3 月 3 日的 XML 形式的中文镜像文件。该数据包中记录了所有在线维基文本的基本信息，如，文章 ID，文章标题，命名空间，文章内容等。

实验过程中，在自由文本中是以句子为单位进行模板的获取和匹配的。因此，需要对下载的维基语料进行格式清理，并做分句处理。此外，还需要对维基文本进行繁简转换。实验中使用的分词，词性标注和实体识别系统 CipSegSDK 是由东北大学自然语言处理实验室开发的。

从维基百科的XML语料中抽取出人物类文本包含的信息盒。经统计，包含信息盒的人物类

文本13505个，其信息盒模板类共132个，属性总数1405个，出现次数大于100的属性有202个。其中有部分属性虽然属性名不同，但实质表示的是同一种属性（多是大小写的不同，同义或是单词分隔符的不同），如date of birth, birth_date, date_of_birth指的都是出生日期，可以统一合并为date of birth。属性合并后，按出现次数由高到低排列，排在前二十位的属性及其出现次数如表2所示。

表2：人物类维基文本中前二十位属性名及其出现次数

属性名	出现次数	属性名	出现次数
name	13505	clubs	3382
date of birth	11399	city of birth	3381
image	8031	current club	3379
place of birth	7209	caps(goals)	3325
date of death	5417	national team	3071
place of death	5107	national years	3065
position	4206	spouse	3050
height	3898	caption	3018
country of birth	3424	youth years	2917
years	3398	youth clubs	2915

4.2 系统评价指标

对特定的属性关系，用算法 2 得到的模板集 P 中的每个模板到测试文本中去匹配，将在文本中用模板匹配得到的三元组和从信息盒中抽取出的三元组进行比较，对系统的性能做出评价。本文用准确率、召回率和 F 值作为系统的评价指标。其中，

$$\mathrm{Precision}=\frac{\text{用模板抽取出的正确的三元组的个数}}{\text{用模板抽取出的三元组的总个数}} \tag{1}$$

$$\mathrm{Recall}=\frac{\text{用模板抽取出的正确的三元组的个数}}{\text{从信息盒抽取出的三元组的总个数}} \tag{2}$$

$$F-Score=\frac{2*\mathrm{Precision}*\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{3}$$

在上述的公式（1）（2）中，对用模板抽取出的三元组的正确性判断，是通过与从信息盒中抽取出的三元组的比较得出的。在比较的过程中，人物姓名和属性名采取完全匹配的方法。在人物姓名和属性名相同的情况下，进行属性值的比较。属性值相似度的计算方法如下：

$$similarity(val1,val2)=0.5*\frac{val1\text{与}val2\text{匹配的片段数}}{val1\text{总的片段数}}+0.5*\frac{val1\text{与}val2\text{匹配的片段数}}{val2\text{总的片段数}} \tag{4}$$

其中，val1 为用模板抽取出来的属性值，val2 为从信息盒中抽取出来的属性值，当 similarity 大于阈值 α 时，则认为抽取出来的属性值正确。

4.3 实验设置及实验结果分析

对实验中用到的六个属性，剔除了属性值为空或是含“不明”、“不详”字样的元组，以确

保有意义的属性值存在。在包含指定属性值的文本中，随机选取 20%作为测试数据，其余的 80%作为测试数据进行了实验。实验中取 m=3，n=3，τ=70%，α=50%，实验结果如表 3 所示。

表 3：系统对不同属性关系的测试结果

属 性 关 系	文本 总数	训练集 文本数	测试集 文本数	模板集 P 大小	准确率 (%)	召回率 (%)	F 值 (%)
date of birth	11168	2234	8934	82	89.7	84.0	86.8
place of birth	5778	1156	4622	101	78.2	77.1	77.6
date of death	3485	697	2788	78	87.5	83.9	85.7
place of death	2695	539	2156	78	77.4	76.0	76.7
spouse	1377	276	1101	54	82.3	80.2	81.2
native place	1024	205	819	47	81.1	79.0	80.0

实验结果表明，在六组实验中，date of birth 和 date of death 这两个属性的准确率都达到了 87%以上，召回率达到了 83%以上。其余四个属性的准确率和召回率都在 75%以上。这说明利用本文中的方法抽取人物属性是可行的。

通过对六组实验结果的分析，得出：

1）影响准确率性能的原因是模板的泛化技术需要进一步改善。实验中综合考虑了准确率和召回率，如果模板泛化程度低一些，准确率会上升，但同时召回率也会有一定程度的下降。

2）影响召回率性能的主要原因是，有些属性值出现的句子并没有出现人物姓名，而是用一些词指代。本实验中没有考虑句子间的联系以及指代消解的问题。这将在下一步的工作中讨论。

5 结论及未来工作

本文提出了一种抽取人物属性的方法。即利用人物类维基文本信息盒中“人物姓名-属性-值”三元组关系，将属性对应的人物姓名和属性值标记到维基文本对应的句子中，用标记好的句子作为训练数据，通过机器学习自动生成模板，从而实现信息盒中人物属性的自动抽取。该方法利用了人物类维基文本中已有的信息盒，可以在信息盒属性值不完善或是信息盒缺失的情况下提取出人物的重要属性，对于抽取大量的人物属性具有重要意义。最后，在人物类维基文本的数据集上进行了实验。实验结果表明，此方法对于抽取维基文本中的人物属性具有较好的效果。

下一步的工作：1）在句子中标记人物姓名时，使用指代消解技术，从而标记到更多的句子，生成更多的模板；2）采用更好的自动生成模板的方法，提高系统的性能；3）用生成的模板集到其他 web 自由文本中去抽取人物属性。

参 考 文 献

[1] 叶正,林鸿飞,苏绥,刘菁菁.基于支持向量机的人物属性抽取.计算机研究与发展.2007(44):271-275.

[2] 王颖.应用于中文人名搜索引擎的 Web 信息提取技术研究.兰州大学硕士学位论文.2006.

[3] Cesar de Pablo-Sanchez and Paloma Martinez.UC3M at WePS2-AE: Acquiring Patterns for People Attribute Extraction from Webpages. In Proceedings of WWW 2009, 2009.

[4] 于满泉.面向人物追踪的知识挖掘研究.中国科学院计算技术研究所博士学位论文.2006.

[5] Deepak Ravichandran and Eduard Hovy.Learning Surface Text Patterns for a Question Answering System. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002:41-47.

[6] F.Wu and D.Weld. Autonomously semantifying Wikipedia. In Proceedings CIKM 07, 2007.