基于网页中深度并列结构的实例提取算法*

张星星，穗志方
北京大学 计算语言研究所，北京 100871
E-mail: zhangxingxing@pku.edu.cn; szf@pku.edu.cn

摘　要：本文发现了网页文件中一种普遍存在的描述性结构—深度并列结构，并使用它来进行概念实例提取。首先提取网页文件中的深度并列结构，用种子实例对其进行过滤和提取候选实例；在候选实例评价阶段，构造种子、网页文件、并列结构和候选实例之间的关系图，并使用 PageRank 算法评价候选实例。在提取的 8 个概念中平均准确率达到 98.25%，平均召回率达到 77.26%。比 R.C.Wang 的提取结果有较为明显的提升。

关键词：深度并列结构；概念实例提取；HTML 标签；PageRank 算法

An Instance Extraction Algorithm Based on Deep Parallel Structures in Web Pages

Zhang Xingxing, Sui Zhifang
Institute of Computational Linguistics, Peking University, Beijing 100871
E-mail: zhangxingxing@pku.edu.cn; szf@pku.edu.cn

Abstract: In this paper, we find a kind of descriptive structure,the deep parallel structure, which is universally exists in web pages. And it was used for the concept instance extraction. At first, we extract the deep parallel structures in web pages, and use the seed instances to filter them and extract the candidates; during the stage of evaluating the candidates, we build a graph among the seeds, web pages, parallel structures and candidates, and use PageRank algorithm to evaluate the candidates. We got the average precision 98.25% and average recall 77.26% of 8 concepts, higher than R. C. Wang's results.

Keywords: deep parallel structure; concept instance extraction; HTML tags; PageRank

1 引言

概念实例提取是找出一个给定概念的所有实例，对一些 NLP 任务（如问答系统[1]）有重要作用。搜索引擎公司收集大量实例集合[2]来改善查询建议[3]和分析查询意图[4]。人工构建实例集合很耗费人力，研究者开始研究自动提取实例集合。许多实例提取算法先给定种子，然后通过扩展种子进行提取。这些方法有基于机器学习的（[5]），但很多是基于模式的，使用人工模式（如[6]、[7]），或通过种子实例获取模式，[8]利用概念和种子实例构造模糊搜索引擎查询，利用种子评价模式；[2]利用种子在搜索引擎查询日志中的前后缀做为模式；[9]利用种子在网页上的最大公共前缀和后缀作为模式；[10]利用网页和普通文本中出现的并列结构进行扩展。

上述研究中[6]、[8]、[2]利用普通文字层面的模式，利用的语料库是普通文本。Web 上可获得大量的网页，它们都带有标签。[10]和[9]利用了网页中表示并列的标签，但[10]仅仅利用了最基本的列表
…，表格等并列结构，且只有在
…
（或者其他并列标签）之间的字符串是种子时才能扩展种子实例，如在图 1 中
…
之间是很多标签，好像没有并列的内容，而在…之间的字符串“让子弹飞”和其他的
…
之间的对应位置的字符串“观音山”、“最强囍事”、“创战纪”在结构和语义上是并列的。这种带嵌套的并列结构在网上大量存在。[9]试图利用标签，但提取模板时从字符串层面寻找各个种子的最大公共前后缀，没有把标签看成整体，没有很好的利用网页结构化的特性。基于以上分析，本文提出基于网页中深度并列结构的实例提取算法。利用深度并列结构提取候选实例。

* 本文相关研究得到国家自然科学基金 60873156、61075067 以及国家社会科学基金 09BYY032 的支持

```
<ul>
    <li>
        <p><img src="rzdf.jpg"></p>
        <p>片名：<a style = "front-size:14px" href="rzdf.htm">让子弹飞</a></p>
        <p>评分：<em>9.6</em></p>
    </li>
    <li>
        <p><img src="gys.jpg"></p>
        <p>片名：<a style = "front-size:14px" href="gys.htm">观音山</a></p>
        <p>评分：<em>6.0</em></p>
    </li>
    <li>
        <p><img src="zqxs.jpg"></p>
        <p>片名：<a style = "front-size:14px" href="zqxs.htm">最强囍事</a></p>
        <p>评分：<em>6.0</em></p>
    </li>
    <li>
        <p><img src="czj.jpg"></p>
        <p>片名：<a style = "front-size:14px" href="czj.htm">创战纪</a></p>
        <p>评分：<em>8.1</em></p>
    </li>
</ul>
```

图 1　含有并列结构的网页示例

2　基本思想

Web 上有很多并列结构，如电影网站上会有很多图文相间的结构来描述电影，往往列出电影的海报、名称等信息（见图 1），这些并列的结构很多描述的是同一类型事物（如图 1 都是电影），使用图文相间结构描述事物的现象在网上很常见。本文观察到若一些实例以上述并列结构在网上出现，它们并列部分对应位置的标签是完全相同或者大部分位置相同的。

本文把表示并列的 html 标签（如
T
…、<table><tr><td>T</td></tr>…</table>等）称为基本并列结构，在基本并列结构的每个并列项中可以嵌套任意数量的 html 标签，如果任何一个并列项中对应位置的标签名称是相同的（标签的属性和属性值可以不同，标签之间的内容可以不同），这样的并列结构是深度并列结构。如图 1 中的列表就是深度并列结构。

深度并列结构要求并列项对应位置标签名称相同，忽略标签属性和属性值，两个标签之间的内容可以不同，避免了因候选实例跟种子周围字符串不一样而被筛掉，比较好的利用了网页结构化的特性。能发现潜在并列实例，如图 1 中的“让子弹飞”、“观音山”、“最强囍事”、“创战纪”。

实例提取系统可分为三个模块：语料获取、实例提取和候选实例评价模块。系统输入为种子实例和与概念相关的关键字。语料获取模块通过给搜索引擎提供与提取概念相关的关键字，取搜索结果的前 150 条，并下载相应的页面。例如如果要提取最新电影，可以给搜索引擎提供关键字“最新电影”。实例提取模块通过种子实例利用深度并列结构提取候选实例。候选实例评价模块建立图结构，并通过 PageRank 算法计算图中每个顶点的权重评价候选。

3　候选实例提取

3.1　自动提取网页中的并列结构

由深度并列结构的定义知其并列项对应位置标签相同，则并列结构可用一串标签和重复次数

表示。首先把网页变成标签序列，记录每标签的名称和它在网页中的位置（为了让标签和网页的内容对应起来）。然后从标签序列中找出所有连续重复的子序列[1]，本文的实验中只考虑重复体长度在 4~100 之间，重复次数大于 3 的子序列。这样找出的深度并列结构有一些重复，如序列
<a>
重复出现了 50 次，那么就会得到长度为 4 重复 50 次、长度为 8 重复 25 次、…、长度为 48 重复 4 次的并列结构，这些并列结构实质上表示的是同一个并列结构，只是跨度不一样。还需要对前两步提取出的并列结构进行提纯，同一位置开始的并列结构只保留重复体最短的。

3.2 提取深度并列结构模板及候选实例提取

使用种子实例对并列结构过滤。遍历所有在 3.1 中找到的并列结构，找出至少一个种子实例出现在并列结构两标签之间的并列结构，记录种子实例在其重复体中出现的位置（即种子在重复体的第几个标签后面出现）。对满足上述条件的并列结构找出它所有重复体种子出现位置的字符串作为候选实例。图 1 网页使用 3.1 和 3.2 中的实例提取算法，得到的实例提取结果如表 1 所示。

表 1　种子实例为“让子弹飞”对图 1 中网页的提取结果

并列结构	重复体	种子位置	重复次数
	<img> <a> 	6	4
候选实例	让子弹飞、观音山、最强囍事、创战纪		

4 候选实例评价

深度并列结构提取出的候选实例中难免有错误实例，如可能由于页面布局原因，一些名字比较长的电影如“将爱情进行到底”显示为“将爱情进行”。因此需要对候选实例评价。本文利用实例提取过程中的信息构造种子、网页、并列结构和候选间的关系图，并用 PageRank 评价候选。

4.1 构建融合深度并列结构的实例提取关系图

[9]在评价候选实例时建立一个图结构，本文的图结构与之类似，只是 Wang 在建立图结构时的模板（Wrapper）顶点被替换为并列结构。直觉上好的并列结构提取出的候选实例比较好，能提取出好候选的并列结构比较好；同样，好的网页能够提取出的并列结构比较好，能够提取出好并列结构的网页比较好。本文建立提取关系图来表达上述关系，把种子实例、网页、深度并列结构和候选作为图顶点。若一个网页 w 提取出了一个深度并列结构 p，则建立从 w 到 p 的双向边；若一个深度并列结构 p 可以提取出一个候选实例 c，则建立 p 到 c 的双向边。在提取候选阶段，种子发现了能够提取出含有种子的深度并列结构的网页，所以建立种子和这些网页之间的双向边。建立双向边为了让网页和并列结构、并列结构和候选及种子和网页之间可以相互评价。

Page et al.提出的 PageRank 算法[11]用来评价网页的权重，核心思想是：在评价网页时不同质量网页对它指向的网页的贡献大小是不一样的。本文使用 PageRank 算法来评价上述图中顶点的权重，核心思想是：在评价候选实例时不同质量网页、并列结构和实例对其相邻顶点的贡献不同。图 2 为一个实例提取关系图的示例。“火箭 湖人 热火”是种子实例，“sports.sohu.com”和“china.nba.com”是两个网

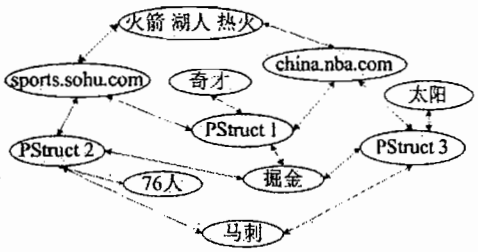

图 2　实例提取关系图的示例

[1] 一个序列 ABABCABCD 中的连续重复子序列为 ABAB 和 ABCABC。

页，PStruct1、PStruct2 和 PStruct3 是并列结构，其他顶点为候选实例。

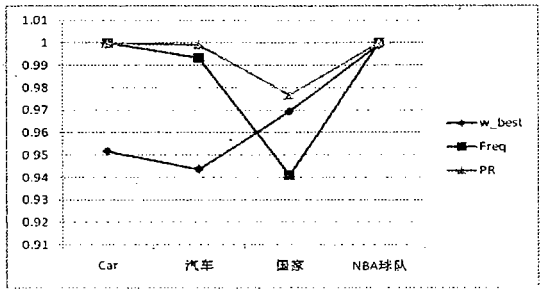

图 3　与 Wang 的算法的比较

4.2　PageRank 评价候选实例

实例的评价使用 PageRank 算法。在建立的实例提取关系图 G 中

$$\forall < u,v > \in E, w(< u,v >) = 1 \tag{1}$$

其中，$w(<u,v>)$为边$<u,v>$的权重。PageRank 算法中的转移概率矩阵 M 为

$$M_{x,y} = \frac{(1-\lambda) * w(< x,y >)}{\sum_{<x,u> \in E} w(x,u)} + \frac{\lambda}{|V|} \tag{2}$$

在实验中，λ取 0.15。使用 v_t 表示每个顶点第 t 次迭代的概率值，则

$$v_{t+1}{}^T = v_t{}^T M \tag{3}$$

v_0 的值为每个顶点最初的概率分布，在 PageRank 中假设图中每个顶点最初概率分布为 $1/|V|$。PageRank 收敛后，对 G 中每个顶点按照 v_t 中每一维进行排序作为最终的候选实例排序。

表 2　语料库的构建

概念名	关键字和下载网页数量	语言
电影	最新电影@150	中文
歌曲	最新流行歌曲@150	中文
歌手		
NBA 球队	NBA@150	中文
汽车	汽车 报价@150	中文
笔记本电脑品牌	笔记本 报价@150	中文
国家	奥运金牌榜@50 世界杯@50 欧洲杯@50	中文
Car	car price@150	英文

5　实验

5.1　语料库的构建

实验过程中首先使用一些与提取实例主题相关的关键字在搜索引擎中搜索，下载前 150 篇文档作为语料。对中文概念抓取 Baidu 的搜索结果，对英文抓取 Yahoo!的搜索结果。不同的概念在构建语料库时使用查询关键字如表 2 所示。

5.2　实例提取及评价

在实例提取过程中，并列结构中重复体长度并非越长越好，重复体长度超过 100 的并列结构

经观察发现多数跨越多个表格。如一个 NBA 球员的列表和一个 NBA 球队的列表，还有其他两个列表，如果这 4 个列表结构完全相同，那么提取出来的并列结构可能会跨越这 4 个表格，而提取的候选实例是这 4 个表格对应位置的内容，产生错误。所以限制重复体的最长为 100。

实验中除了使用 PageRank 算法进行评价，还使用了一种简单的评价方法，即按照一个候选实例被提取出的频次进行评价。在评价提取结果时使用准确率（Precision）、召回率（Recall）和平均准确率（MAP）。平均准确率（AP）可以比较好的反映一个排序的质量，平均准确率（MAP）是几次排序结果的平均准确率（AP）的平均值。

$$AP(L) = \frac{\sum_{r=1}^{|L|} Prec(r) * CorrectInstance(r)}{\#CorrectInstances} \tag{4}$$

L 是候选实例的排序，$Prec(r)$是前 r 个实例的准确率，$CorrectInstance(r)$当第 r 个实例为正确的实例时它返回 1，反之返回 0。实验中对每个概念使用三组种子进行提取，计算 MAP。

5.3 实验结果

表 3 中是本文实验结果。Freq 表示基于频次的方法评价候选实例，而 PageRank 表示 5.1 和 5.2 的方法评价候选实例。概念“电影”、“歌曲”和“歌手”没有评价召回率，因为这 3 个概念变化太快，很难找到一个黄金标准。对“NBA 球队”，使用 NBA 官网提供的 30 支球队作为黄金标准；对“汽车”使用汽车之家列出的 101 种品牌为黄金标准；对于“笔记本”，使用太平洋电脑网提供的 26 个主流笔记本品牌作为黄金标准。对“国家”使用百度百科提供的 194 个国家作为黄金标准；对“Car”使用[9]提供的“popular car maker”的 56 个汽车品牌作为黄金标准。

实验结果表明，利用 PageRank 算法进行评价的结果多数情况下平均准确率、召回率和 MAP 大于使用频次的方法。使用 PageRank 算法评价候选实例的 8 个概念平均准确率高达 98.25%，使用基于频次的方法也可以达到平均 93.47%的准确率。而两种评测方法的召回率都在 75%左右。

表 3 实验结果

	数量	Precision		Recall		MAP	
		Freq	PageRank	Freq	PageRank	Freq	PageRank
电影	300	95.00%	97.00%			98.08%	99.20%
歌曲	100	84.00%	99.00%			90.96%	99.85%
歌手	100	99.00%	99.00%			99.95%	99.87%
NBA 球队	30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
汽车	100	98.00%	98.00%	74.26% 90.10%@197	76.24%	99.33%	99.92%
笔记本	35	85.71%	100.00%	69.23%	73.08%	95.56%	100.00%
国家	100	87.00%	94.00%	43.59% 50.77%@205	47.69%	94.09%	97.67%
Car	101	99.01%	99.01%	89.28%	89.28%	100.00%	99.96%
平均值	108.25	93.47%	98.25%	75.27%	77.26%	97.25%	99.56%

5.4 与已有算法的比较

Wang 在[9]中测试了 36 个概念（英文、中文和日文语料各 12），所有提取结果只给出了 MAP 的评测结果，且并没有说明对每个概念评价时实例的具体数量，仅给出了上限。本文提取的概念中有 4 个与 Wang 的相同的，对 Wang 实验结果取上限为 100 的 MAP 值。从实验的结果（图 3）可

以看出，本文的实例提取算法用 PageRank 对候选实例进行评测的 4 个类中，除了 NBA 球队和 Wang 的都是 100%，其他结果高于 Wang。

5.5 本文算法的适用范围

本文算法的原理是利用网页中的深度并列结构进行实例提取。算法对常出现在网页并列结构中的实例提取效果较好。从表 3 可以看出概念“国家”的准确率和召回率最低。因为很少有人把“国家”的实例信息在网页上列出来，为提取“国家”的实例，本文使用三组关键字获取语料。对网民们关注的概念，如“电影”等，很多网站收集这些实例信息，并把它们以结构化的形式呈现在网上，这些结构很多是深度并列结构。购物网站也常把商品以结构化的形式放在网上。所以本文的实例提取算法对于网民关注的概念，及一些网站提供的商品概念有较好提取效果。

6 结论

本文发现了网页中普遍存在的深度并列结构，并用它进行实例提取，获得较好提取效果。本文算法对网民们比较关注的概念，如“电影”、“歌曲”、“NBA”等，及一些购物网站提供的商品相关的概念，如笔记本品牌、汽车品牌、服装品牌等有较好提取效果。此外，一些实例及其属性会同时出现在深度并列结构中，下一步，我们将尝试利用深度并列结构同时提取实例和属性。

参 考 文 献

[1] Richard C. Wang, Nico Schlaefer, William W. Cohen, and Eric Nyberg. Automatic set expansion for list question answering[C]. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Hawaii, October 2008: 947-954.

[2] Marius Pasca. Weakly-supervised discovery of named entities using web search queries[C].Proceedings of the 16 ACM conference on Conference on information and knowledge management, New York,2007: 683-690.

[3] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data[C]. In Proceedings of KDD-08, 2008: 875-883.

[4] J. Hu, G. Wang, F. Lochovsky, J. tao Sun, and Z. Chen. Understanding user's query intent with Wikipedia[C]. In Proceedings of WWW-09, 2009: 471-480.

[5] Huang, R. and Riloff, E. Inducing Domain-specific Semantic Class Taggers from (Almost) Nothing[C]. Proceedings of The 48th Annual Meeting of the Association for Computational Linguistics, 2010.

[6] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy.Semantic class learning from the web with hyponym pattern linkage graphs[C]. In Proceedings of ACL-08: HLT, Columbus, Ohio, June 2008: 1048-1056.

[7] Sarmento, L.; Jijkuon, V.; de Rijke, M.; and Oliveira, E. "More like these": growing entity classes from seeds[C]. In Proceedings of CIKM-07, Lisbon, Portugal, 2007: 959-962.

[8] Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations[C]. In Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics, Sydney, Australia, 2006: 113-120.

[9] Richard C. Wang and William W. Cohen. 2007.Language-independent set expansion of named entities using the web[C]. In ICDM, 2007: 342-350.

[10] Shuming Shi, Huibin Zhang, Xiaojie Yuan, and Ji-Rong Wen. Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches[C]. In the 23rd International Conference on Computational Linguistics,Beijing, August 2010.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web[R]. Technical report, Stanford Digital Library Tech. Project, 1998.

基于电子商务用户行为的同义词识别*

张书娟，董喜双，关 毅
哈尔滨工业大学，哈尔滨 150001
E-mail: zhangsj777@gmail.com; dongxishuang@gmail.com; guanyi@hit.edu.cn

摘 要：本文研究了电子商务领域同义词的自动识别问题。针对该领域新词多、错别字多、近义词多的用词特点，提出基于用户行为的同义词识别方法。首先通过并列关系符号切分商品标题和基于 SimRank 思想聚集查询两种方法获取候选集合，进而获取两词的字面特征以及标题、查询、点击等用户行为特征，然后借助 Gradient Boosting Decision Tree（GBDT）模型判断是否同义。实验表明同义词识别准确率达到了 54.46%，高于 SVM 近 4 个百分点。

关键词：同义词识别；用户行为；SimRank；Gradient Boosting Decision Tree

The Synonym Recognition Based on User Behaviors in E-commerce

Zhang Shu-juan, Dong Xi-shuang, Guan Yi
Harbin Institute of Technology, Harbin 150001
E-mail: zhangsj777@gmail.com; dongxishuang@gmail.com; guanyi@hit.edu.cn

Abstract: This paper focuses on synonym recognition in e-commerce. Considering there are more new words, typos, and near-synonyms in e-commerce domain, we present a method to recognize synonyms based on user behaviors. Firstly, candidate sets are retrieved by analyzing the titles and their corresponding queries based on SimRank theory, and then, features including literal feature, title feature, query feature, click feature are extracted. Finally, Gradient Boosting Decision Tree(GBDT) model is adopted to determine whether candidate synonyms are real synonyms or not. The experimental result shows that GBDT model is more suitable for this task with precision 54.46%, which is nearly 4 percent higher than that of SVM.

Keywords: synonym recognition; user behaviors; SimRank; Gradient Boosting Decision Tree

1 引言

随着互联网的发展，电子商务逐步发展起来。对于电子商务网站的站内搜索引擎而言，应该在准确理解买方意图的基础上，尽可能多的检索出相关商品，所以需要对查询进行扩展。要准确扩展查询，同义词表是必须的也是最基础的资源。目前国内还不具备电子商务领域的同义词典，而手工构建又费时费力，所以需要采用同义词自动识别的方法。

牛津字典对同义词的定义为：在用同一种语言表达的意义相同或者相近的两个词或者短语[1]。而电子商务中则要求意义完全相同，定义为对同一事物或者概念的不同表达，即在商品检索和商品描述中可以互相替换的词[2]。电子商务领域中同义词主要有六类：（1）中英文名称，如：耐克-Nike。（2）学名与俗名，如：圣女果-小番茄。（3）全称与简称，如：美特斯邦威-美邦。（4）新称与旧称，如：自行车-脚踏车。（5）常用错别字引起的同义，如：瑜伽-瑜珈。（6）传统同义词，如：储物柜-收纳柜。

电子商务领域同义词的特殊性，使现有自动识别方法的效果大打折扣。一方面现有资源中给出的同义词不一定满足此领域中的定义，例如在《同义词词林》中木耳与黑木耳为同义词，其实两者是上下位关系，木耳除了包括黑木耳之外还有秋木耳、白木耳等。另一方面网络用词中新词多、错别字多，大量的词现有资源还未收录。因此，需要根据电子商务领域数据特点寻找新的同义词的识别方法。

* 本项研究受到国家自然科学基金项目支持，项目批准号：60975077，60736044。