面向句法结构的文本检索方法研究

马路遥,夏 博,肖 叶,荀恩东
(北京语言大学信息科学学院,北京 100083)

摘 要: 语言资源加工和语言学研究,对大规模树库的结构化检索有很高需求. 本文针对句法树语料设计了索引、检索方法. 针对汉语的特点以及知识抽取任务的需求,我们设计了七种索引结构,旨在借助句法树的结构、属性信息,进行高效、准确的知识抽取. 本方法不仅支持字符串检索、属性检索,也支持基于句法树结构、属性信息的检索. 实验证明,本方法高效、准确.

关键词: 句法树语料; 知识抽取; 信息检索; 语言资源
中图分类号: TP391.12 **文献标识码:** A **文章编号:** 0372-2112 (2020)05-0833-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.05.001

Structural Retrieval on Chinese Syntax Tree Corpus

MA Lu-yao, XIA Bo, XIAO Ye, XUN En-dong
(*Beijing Language and Culture University, School of Information Science, Beijing* 100083, *China*)

Abstract: Language resource processing and linguistics research require effective retrieval on syntax tree corpus. This paper presented an index and search method for syntax tree corpus, which is efficient, accurate, and flexible. Based on the features of Chinese language and the needs for knowledge extraction, we designed seven types of indexes, aiming that with the help of structure and attribute information, knowledge extraction will be performed more effectively and accurately. Apart from general retrieval functions, our method supports retrieval based on the structure and attribute information of syntax trees. Experiments show that our method is both accurate and efficient.

Key words: syntax tree corpus; knowledge extraction; information retrieval; language resource

1 引言

基于句法结构的知识抽取对于计算语言学和语言学研究都具有重要的实践意义. 面对海量规模语言资源,尤其是深度加工的句法结构资源,准确、高效的检索系统是十分重要的. 现有语料库工具并不能满足知识抽取的需求,原因如下:(1)检索功能有限;(2)无法处理大规模语料;(3)检索过程是形式上的符号匹配,不能处理歧义结构. 因此,我们从数据和算法的角度出发,提出了面向句法树语料的可编程检索方法.

本方法可以根据汉语的特点,借助句法树的属性(树节点标签)、结构信息(层次结构),建立索引,从而提高检索结果的准确性. 比如,在线性语料中检索框式结构或离合结构时,由于难以限定词与词之间的距离,结果中将存在大量歧义结构. 而句法树的结构、属性信息则可以对词之间的距离给出自然的限制,从而排除歧义结构. 此外,在实现上,本方法使用了高效的数据结构和算法,保证了较高的效率.

2 相关工作

当前主流的语料库多采用生语料或者分词词性标注语料(以下称线性语料),语料本身无结构信息,如北京大学CCL语料库[1]、国家语委现代汉语通用平衡语料[2]、BCC语料库[3]、DCC动态流通语料库[4]、台湾中研院现代汉语平衡语料库[5]等. 从语料规模以及检索效果来看,现有语料库都无法满足知识抽取的需求. 此外,深度学习技术[6,7]的发展使得获取大规模标注句法树语料成为可能. 因此,我们希望借助句法树语料的结构、属性信息,发挥数据规模的优势,进行准确、高效的知识抽取.

收稿日期:2019-09-10;修回日期:2020-01-06;责任编辑:马兰英
基金项目:国家社会科学基金重点项目(No. 16AYY007);北京市语言资源高精尖创新中心科研项目(No. TYR17001J);北京语言大学研究生创新基金(No. 19YCX118)

目前已提出了一些句法树检索方法,如针对德语树库的 TIGERSearch[8],针对宾州树库的 Tgrep2[9] 和 Tregex[10]. 2017 年 Juhani 等在 SETS[11] 依存树检索工具的基础上提出了时空效率更高的 Dep Search[12],但是 20000000 句树语料生成的索引数据仍有 45G 之多. 此外,也有文章对已有树检索方法进行了审视,提出了检索语言应满足的要求[13,14]. 上述方法的检索语言多以树节点为检索单位,检索式多用于描述子树的模式,而且检索功能有限,输出结果不便于后续统计分析,并且不能处理大规模数据. 本方法针对汉语的特点以及知识抽取的需求设计了基于句法树的索引结构.

3 索引的设计与实现

3.1 检索语言

对比于其他树检索语言,本方法的检索语言表达能力更强,更加灵活. 检索符号如表 1 所示. 每种检索类型对应特定模式的检索式,详见 3.3 节.

表 1 检索符号

类型	说明与举例
汉字串	汉字串,不限长度.
属性符号	句法树节点的属性标签.
边界符号	e 表示边界,bTag 左边界,eTag 右边界. 其中 Tag 是具体的短语标签.
离合符号	* 小句内离合,^小句间离合.
通配符	“ ~ ”表示一个词,“. ”表示一个字符.
( )	括号用于指定输出内容,便于统计分析. 括号内的内容用“ $ ”+括号序号表示,例如 $1, $2 等.

3.2 索引原理

3.2.1 倒排索引

倒排索引[15,16] 是搜索引擎中常用的索引结构,建立了单词与该单词在文档的位置之间的映射,一个单词对应一个由倒排项组成的倒排列表. 如图 1 所示.

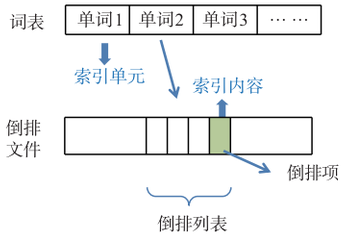


图1 倒排索引相关概念示意图

本方法设计了从索引单元到索引项的倒排索引. 索引项在建索引时是指用于计算索引内容的节点. 在检索时,是指从原子检索式中解析出的、用于与索引数据比较的内容. 索引单元可以是句法树节点的内容、属性或者边界信息. 索引内容保存了索引项所在句子在语料中的偏移量、索引项在句中的位置、以及索引单元的相关信息. 在检索时,通过索引内容可定位索引项在语料中的位置,还原出语料内容,从而进行比较和查找.

3.2.2 索引内容计算

保存的句法树结构、属性信息如图 2 所示. 同时,以词(叶节点内容)为单位将句子转换为一个 ID 序列,利用 ID 序列计算句子偏移量以及索引项的位置. 例,原句开始于全文的第 50 个词,那么句子偏移量为 50. 索引项的位置则是节点在句内的开始位置,如索引项是“书籍”,对应 ID 序列中的第 4 个,则其位置是 4. 将语料转化为 ID 流简化了计算过程,并且节省空间. ID 与词之间的转换是通过一个全局词典来实现的.

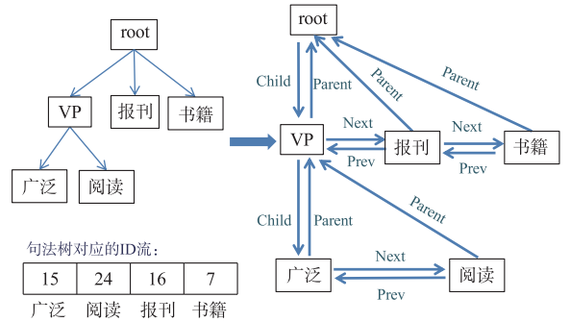


图2 保存的句法树信息

3.3 索引设计

本方法设计了 4 种检索类型,定义了 7 种索引,如表 2 所示. 其中,“HZ”表示一个汉字,“Tag”表示树节点标签,“CI”表示一个词(叶节点),“string”表示汉字串. 每种检索类型对应特定的原子检索式模式,不同原子检索式之间可以组合,构成复杂检索式.

表 2 索引设计

检索类型	原子检索式模式	索引单元	索引项	检索式举例
字串检索	HZ string	HZ	紧邻后缀	锻炼身体
属性检索	string Tag	Tag	紧邻前缀	打击 Np
	Tag string	Tag	紧邻后缀	Ap 的记忆
结构检索	CI * string e	CI	离合后缀	用 * 吃饭 e
	string e * CI	CI	离合前缀	打击 e * 敌人
语块检索	bTag * string eTag	ML	离合后缀	bNp * 灯 eNp
	bTag string * eTag	MR	离合前缀	bVp 打击 * eVp

(1)字串检索

索引单元是词首汉字,索引项是该词后紧邻的词. 该设计实现了全文检索功能.

(2)属性检索

索引单元为句法树节点的属性标签. 索引项为索

引单元前、后的词. 该索引类型支持短语级的属性查询. 举例:打击 Np. 检索式表示“打击”后跟名词短语 Np.

(3)结构检索

该索引具有良好的可扩展性,可以根据语言现象的特点,借助结构、属性信息建立索引. 以抽取介词-动词搭配为例,我们以介词为索引单元,借助句法树的结构、属性信息选择来作为索引项的动词,如图 3 所示. 详见 5.3.2 节.

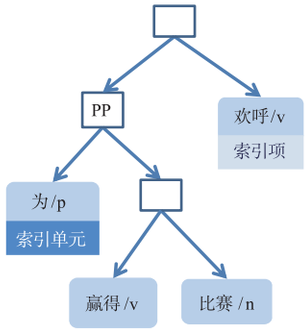


图 3　索引项是动词“欢呼”而不是“赢得”

(4)语块检索

索引单元是表示语块左、右边界的符号,索引项分别是语块的终结叶节点和开始叶节点. 利用该类索引可以抽取带有某个语缀或者具有特定中心词的语块. 举例:bNp * 灯 eNp. 检索式表示以“灯”结尾的名词性短语. 如图 4 所示.

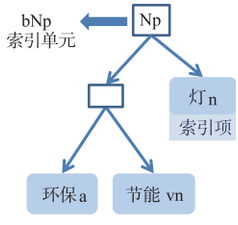


图4　“bTag * string eTag”索引类型示意图

3.4　索引流程

本文采用遍历两遍语料的方法建立索引,如图 5 所示,主要步骤如下:

(1)第一遍扫描语料,计算索引内容所需空间,导出所需数据.

(2)第二遍扫描语料,计算索引内容,填充倒排列表.

(3)对每种索引类型下的索引内容排序.

一个索引单元对应的索引内容列表为一个排序区间. 如图 6 所示,从索引内容得到索引项的位置,从此位置按照一定方向、窗口大小读取 ID 流,将 ID 流转换为字符串,比较字符串,从而进行排序.

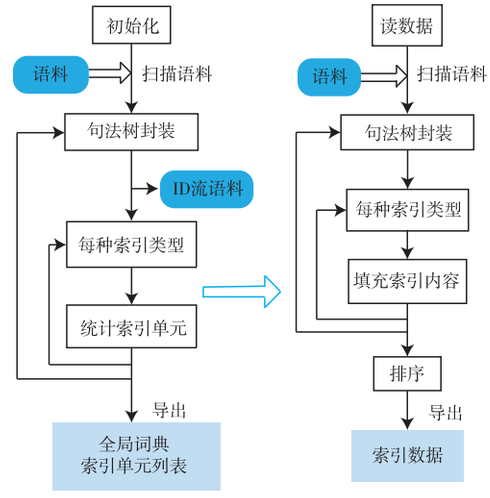


图5　建索引流程

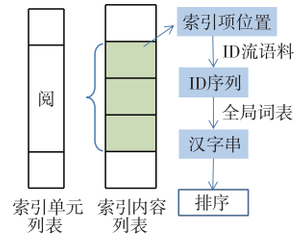


图6　索引内容排序

4　检索算法

4.1　原子查询

检索式会被解析为一个或者多个原子检索式. 一个原子检索式对应一个原子查询. 原子检索式保存了索引类型、索引单元、索引项等信息. 如检索式“阅读书籍”,属于“HZ string”检索类型,索引单元是“阅”,索引项是“读书籍”,查找时使用“读书籍”与索引内容进行比较.

检索时,若索引单元在索引单元列表中,则可得到对应的索引内容区间. 然后通过二分查找,比较索引项与索引内容的对应原文,从而确定结果的上下界. 检索结果是由结果所在句子的偏移量,在句内的位置等来保存的. 如图 7 所示.

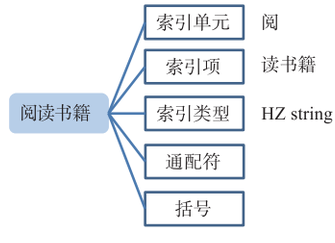


图7　原子检索式解析结果

4.2 搜索流程

检索流程主要包括解析检索式、查询以及合并操作. 解析检索式时,首先按照“^”将检索式分为多个子句,每个子句可以解析出一个或者多个原子查询. 在子句内按解析的顺序查询、合并,然后合并各个子句的查询结果. 判断能否合并的依据是两个结果在原语料中的位置关系. 如图 8、图 9 所示.

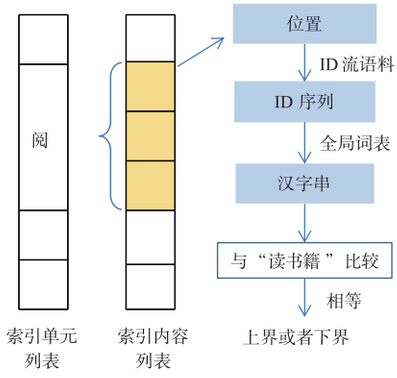


图 8 原子查询示意图

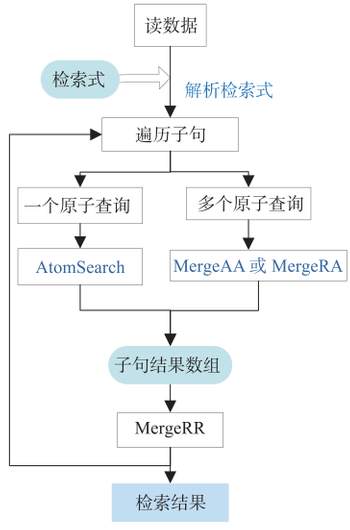


图9 检索流程.Merge-表示合并操作

5 实验

实验分为 3 个部分,第 5.1 和 5.2 节使用大规模句法树进行索引、检索实验,对效率进行考察. 第 5.3 节使用 Chinese Treebank[17] (CTB)标准句法树语料进行实验,对本方法的效果进行评估. 用于实验的机器配置是 Intel(R) Xeon(R) CPU E5-2620V4,内存 96GB.

5.1 索引生成

本实验采用由人民日报语料生成的句法树进行实验. 实验使用了 5 组不同大小的句法树语料,分别为 50MB、100MB、500MB、1GB、3GB. 实验结果如表 3 所示.

从表 3,表 4 可以看出,随着语料增大,建索引所用的时间呈线性增长. 我们可以大概估计建索引所需的时间. 相比于 SETS 和 Dep Search,本方法生成的索引数据所占空间更小. 使用 Dep Search 方法,20000000 句法树对应的索引数据约为 45G.

表 3 索引实验结果

语料大小	用时(min)	索引 (MB)
50MB,164506 句	4.99	144
100MB,289702 句	21.29	383
500MB,1427539 句	108.73	1,741
1GB,3006686 句	256.82	3,754
2GB,5954629 句	548.44	7,324
3GB,9248910 句	875.79	11,112

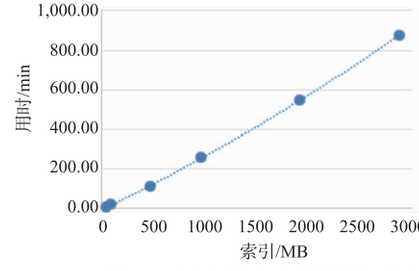


图10 各组语料建索引所用的时间

5.2 检索

用户在 lua 脚本中调用 API,通过命令行运行脚本进行检索. 本小节使用 3GB、9,248,910 句法树语料生成的索引数据,对检索的效率和准确性进行了测试. 检索结果如表 4 所示.

表 4 检索实验结果

检索式	对~的侵略
解析结果	**原子查询** 1 检索类型:HZ string 索引单元:对 索引项:~
	原子查询 2 检索类型:HZ string 索引单元:的 索引项:侵略
用时	读取数据 42.14s 查询合并 45.45s 输出 0.116s
结果	总计 5262 条,部分结果如下: 对越南的侵略 对也门的侵略 对中国的侵略 对东帝汶的侵略 对古巴的侵略……

检索式被解析成了 2 个原子查询,原子检索式 1 表示“对”后跟任意一个词,如“对中国”、“对越南”等. 原子检索式 2 表示字符串“的侵略”. 若两条结果在同一个句子,并且在句中的位置符合检索式的描述,则可以合并. 比如“对越南”和“的侵略”分别对应同一句子的第 5 到第 7 个字和第 8 到第 10 个字,符合检索式的描述,因此可以合并.

查询用时约为 45.5s. 使用 SETS 方法在约 10000000 句语料中检索复杂检索式需要 48s 到 1 分 52s. 本方法在内存中操作,因此更为稳定、高效.

5.3　基于 CTB 数据的实验

5.3.1　检索实验

本文使用了 5.78MB CTB5 语料建立了索引,生成索引数据 21MB. 检索结果如表 5 所示.

表 5　在 CTB 语料上的检索结果

检索式	以 * 为主 e 的 Np
解析结果	**原子查询** 1 检索类型:CI * string e 索引单元:以 索引项:为主
	原子查询 2 检索类型:string TAG 索引单元:Np(名词短语) 索引项:的
用时	读取数据 0.078s 查询合并 0.117s 输出 0.016s
结果	总计 25 条,部分结果如下: 以重化工业为主的产业结构 以技术股为主的纳斯达克 以电子设备厂家为主的日本 以生态环境为主的儿童课辅班 ……

在语料中检索该复杂检索式用时 0.117s. 使用 SETS 在 90K 语料上检索简单检索式,用时为 1.6s. 另外,使用本方法和 Tregex 分别检索“Pp”介词短语,用时分别为 0.22s 和 2.54s. 相比已有方法,本方法具有更高的查询效率. 尽管不同方法的设计原理、检索语言以及所针对的任务不同,我们仍可以从语料规模以及用时上对本方法的效率有一个大致的评估.

信息检索中常用的评价指标是正确率和召回率[16]. 我们主要分析召回率. 针对该检索式,我们用正则表达式抽取了相关原文,人工对比后发现,语料中所有符合的结果均被检索出,因此该检索结果的召回率均达到 100%. 从本方法的原理上看,本方法能够返回所有符合检索式的结果,多数情况下可以保证较高召回率. 但是,结果数目跟参数的设置有关,如最大结果返回数目,索引单元的最大长度等,因此某些情况下召回率有所损失. 比如使用 Tregex 和本方法在 CTB 上分别检索“Pp”介词短语,Tregex 返回了全部结果,共计 13814 条,本方法返回了 13356 条结果,召回率为 96.68%.

抽取的效果取决于检索式的描述能力. 这要求我们对语言现象和语料有充分的了解. 仍以检索“以 * 为主 e 的 Np”为例,句法树结构如图 11 所示,“以包钢稀土研究院为主的世界”与检索式匹配,但是不符合使用习惯. 我们可以从索引设计和检索式两个角度进行优化.

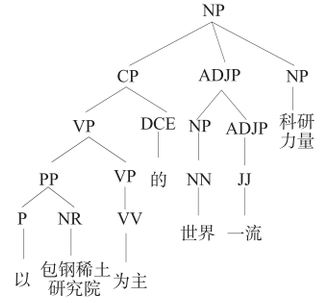


图11　句法树标注实例

5.3.2　搭配抽取实验

以抽取介词-动词搭配为例. 当介词短语修饰其后动词短语时,介词与动词构成搭配. 在线性语料中检索 p * v,表示介词和动词离合出现,符合词性序列的结果都会返回. 例,原文为:

宋浩京/Nr 代表/v 朝方/Nn 对/p 中国/Nr 党政/Nn 领导人/Nn 和/Cc 人民/Nn 哀悼/v 金日成/Nr 主席/Nn 逝世/v 表示/v 深切/Jj 谢意/Nn ./Pu

检索结果有“对 * 哀悼”、“对 * 逝世”和“对 * 表示”. “哀悼”、“逝世”在介词结构内,显然不是介词“对”的搭配. 通过上例可以看出在线性语料中检索的局限性. 利用本方法提高抽取的过程如下:

(1)使用本方法对 CTB 语料建立索引.

(2)检索 p * v. 效果等同于在对应线性语料中检索.

(3)提取、统计候选搭配.

在 7969 组候选搭配中,高频部分的抽取结果大多符合要求,部分抽取结果如下:

以_为 222
在_举行 155
在_说 106
比_增长 99
对_进行 87
为_提供 79
……

我们主要关注低频部分. 其中,频次为 1 的候选搭配有 6194 条,占据结果的大部分.

(4)利用低频(频次为 1)的候选搭配构造检索式,在句法树语料中检索.

例,候选搭配为“对 * 逝世”,对应的检索式为“对 * 逝世 e”.

(5)结果提取.

本实验构造了 6194 条检索式,其中 2351 条无检索结果,于是可以删除. 观察发现,保留的结果绝大部分符合对搭配要求,筛除的部分则不符合要求. 如表 6 所示.

表 6 部分抽取结果

保留的搭配	删除的搭配
针对 问	对 逝世
与 平分秋色	除了 种
从 探访	就 列席
在 富	以 将
以 挥杖	在 深化
……	……

分析:针对该抽取任务建索引时,我们以介词为索引单元,以位于介词短语之外,并且与介词在同一个小句的动词为索引项. 但该限制条件并不能保证抽取的介词与动词构成修饰关系,如图 12 所示. 后续可通过融合语言特点、优化索引设计来进行改进. 综上所述,借助句法结构信息可以提高抽取搭配的效果.

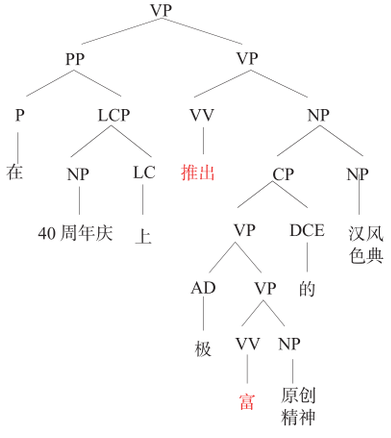


图12 动词“推出”,“富”都会作为索引项

6 结论

本文对面向句法树语料的知识抽取方法进行了探索,建立了基于句法树语料的倒排索引,实现了基于二分查找的检索算法. 本方法支持一般的字串检索、属性检索,还提出了结构检索和语块检索. 相比于已有方法,本方法的检索语言更加直观、灵活,索引设计融合了汉语的特点,考虑了知识抽取的需求. 在实验部分,使用大规模句法树语料和标准句法树语料进行了索引、检索实验. 实验表明,本方法准确、高效.

综上所述,本方法实现了面向语法树语料索引、检索方法,为大规模树结构检索提供了解决方案和工程实践的经验.

参考文献

[1] 俞士汶,段慧明,等. 北京大学现代汉语语料库基本加工规范[J]. 中文信息学报,2002,16(5):49 – 64.

[2] 国家语委语言文字应用研究所计算语言学研究室. 信息处理用现代汉语词类标记集规范[J]. 语言文字应用,2001(3):16 – 20.

[3] 荀恩东,饶高琦,等. 大数据背景下 BCC 语料库的研制[J]. 语料库语言学,2016,3(1):93 – 118.

[4] 国家语言资源监测与研究平面媒体中心. 动态流通语料库使用帮助[Z/OL]. http://dcc. blcu. edu. cn/explain. jsp,2015-05-19.

[5] 黄居人,陈克健. 中央研究院平衡语料库检索系统使用说明[R]. 台湾:中央研究院,1998.

[6] Nikita Kitaev,et al. Constituency parsing with a self-attentive encoder[A]. Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics[C]. Melbourne,Australia: Association for Computational Linguistics,2018. 2676 – 2686.

[7] Daniel Fried,et al. Improving neural parsing by disentangling model combination and reranking effects[A]. Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics [C]. Vancouver,Canada: Association for Computational Linguistics,2017. 161 – 166.

[8] Sabine Brants,et al. TIGER: Linguistic interpretation of a german corpus[J]. Research on Language and Computation,2004,2(4):597 – 620.

[9] L T Rohde,Douglas. Tgrep2 user manual[Z/OL]. https://web. stanford. edu/dept/linguistics/corpora/cas-tut-tgrep. html. 2004.

[10] Roger Levy,et al. Tregex and Tsurgeon: tools for querying and manipulating tree data structures[A]. Proceedings of International Conference on Language Resources and Evaluation[C]. Genoa,Italy: European Language Resources Association. 2006. 2231 – 2234

[11] Juhani Luotolahti,et al. SETS: Scalable andefficient tree search in dependency graphs[A]. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics[C]. Denver,Colorado: Association for Computational Linguistics. 2015. 51 – 55.

[12] Juhani Luotolahti,et al. Dep search: Efficientsearch tool for large dependency parsebanks[A]. Proceedings of the 2017 Nordic Conference of Computational Linguistics [C]. Gothenburg, Sweden: Linkoping University Elec-

tronic Press. 2017. 255 – 258.

[13] Lai, C, Bird, S. Querying and updating treebanks: a critical survey and requirements analysis[A]. Proceedings of the Australasian Language Technology Work-shop[C]. Sydney, Australia: Australasian Language Technology Association. 2004. 139-146.

[14] Jiří Mírovský. PDT 2.0 Requirements on a query language [A]. Proceedings of the 2008 Annual Meeting of the Association for Computational Linguistics: Human Language Technology[C]. Columbus, Ohio, USA: Association for Computational Linguistics. 2008. 37 – 45

[15] Salton Gerard. A Theory of Indexing[M]. Cambridge University Press, 1975. 41 – 48.

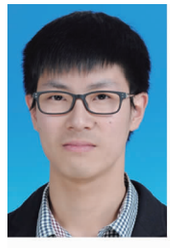
[16] Christopher D. Manning, et al. Anintroduction to information retrieval [M/OL]. https://nlp. stanford. edu/IR-book/, 2009.

[17] Xue Nianwen, et al. The penn chinese treebank: phrase structure annotation of a large corpus[J]. Natural Language Engineering, 2005, 11(2). 207 – 238.

作者简介

马路遥　女,硕士研究生,1994 年出生. 研究方向为自然语言处理.
E-mail:maluyao_blcu@ outlook. com

肖　叶　女,硕士研究生,1996 年出生,研究方向为自然语言处理.
E-mail:blcuxiao@ 126. com



夏　博　男,硕士,1993 年出生,研究方向为计算机网络,自然语言处理.
E-mail:blcuxiabo@ 126. com

荀恩东(通信作者)　男,1967 年出生,博士,教授,主要研究领域为计算语言学、语言教育技术.
E-mail:edxun@ 126. com