中国科学: 信息科学 2017 年 第 47 卷 第 8 期: 1–17
SCIENTIA SINICA Informationis
论文

基于块密度加权标签路径特征的 Web 新闻在线抽取

吴共庆*, 刘鹏程, 胡骏, 胡学钢*

合肥工业大学计算机与信息学院, 合肥 230009
* 通信作者. E-mail: wugq@hfut.edu.cn, jsjxhuxg@hfut.edu.cn

收稿日期: 2017–04–05; 接受日期: 2017–06–08; 网络出版日期: 2017–07–17
国家重点研发计划 (批准号: 2016YFB1000901)、教育部创新团队发展计划 (批准号: IRT13059)、国家自然科学基金 (批准号: 61273297, 61673152) 和国家留学基金 (批准号: 201506695019) 资助项目

摘要 Web 新闻内容抽取是众多“大数据”和“大知识”应用的基础, 也是一个开放性问题. 标签路径特征和文本块密度特征是目前解决该问题的两类优良特征. 标签路径特征能较好地区分全网页的内容与噪音, 但难以识别内容块中的噪音和噪音块中的内容; 文本块密度特征能较好地识别高密度的内容块, 但鲁棒性不足. 因此, 本文提出了一种可有效结合标签路径特征和文本块密度特征的 Web 信息抽取模型 CEDP, 结合两种特征的优点, 设计了一种基于文本块密度加权的标签路径特征, 并设计了基于该特征的 Web 新闻抽取算法 CEDP-NLTD. CEDP-NLTD 是一种快速的、通用的、无需训练的在线 Web 新闻内容抽取算法, 适用于 Web 大数据环境下的多种来源、多种风格、多种语言的异构 Web 新闻网页抽取任务. 在 CleanEval 等测试数据集上的实验结果表明, CEDP-NLTD 方法优于 CETR, CETD, CEPR, CEPF 等在线抽取方法, 且优于基于 CEDP 模型直接使用 CETD 方法设计的 3 种块密度特征所形成的算法 CEDP-TD, CEDP-CTD, CEDP-DSum.

关键词 内容抽取, Web 新闻, 文本块密度, 标签路径特征, 在线算法

1 引言

Web 新闻是一种重要的互联网大数据源. 根据中国互联网络信息中心 (CNNIC) 2016 年 7 月发布的第 38 次《中国互联网络发展状况统计报告》[1], 3 大基础互联网应用 (即时通信、搜索引擎、网络新闻) 的用户规模保持稳健增长; 在 2015.12~2016.6 期间, 网络新闻在中国网民各类互联网应用中的使用率排名第三, 在各类手机互联网应用中的使用率排名第二; 截至 2016 年 6 月, 中国网络新闻用户达到了 5.79 亿, 比 2015 年底规模增加了 1487 万, 网络新闻的使用率为 81.6%; 手机网络新闻用户达到了 5.18 亿, 比 2015 年底规模增加了 3635 万, 手机网络新闻的使用率为 78.9%.

引用格式: 吴共庆, 刘鹏程, 胡骏, 等. 基于块密度加权标签路径特征的 Web 新闻在线抽取. 中国科学: 信息科学, 2017, 47: 1–17, doi: 10.1360/N112016-00305
Wu G Q, Liu P C, Hu J, et al. Online Web news extraction via tag path feature weighted by text block density (in Chinese). Sci Sin Inform, 2017, 47: 1–17, doi: 10.1360/N112016-00305

Web 新闻内容抽取是搜索引擎、信息检索、多源异构动态 Web 信息处理、评论分析与情感计算、个性化推荐服务、高质量网页打印、手持设备阅读、情报获取与安全、舆情监测、网络新闻聚合、股票行情预测、新闻事件分析、话题跟踪等众多 “大数据” 和 “大知识” 应用 [2~4] 的基础技术. Web 新闻网页中不仅包括用户感兴趣的新闻 “内容”, 同时还有导航、广告、推荐链接、版权声明、免责声明等 “噪音” 信息. 这些 “噪音” 信息的存在给很多互联网应用带来障碍. 在网络舆情分析 [5]、新闻事件分析、话题检测与跟踪 [6] 等应用领域, Web 新闻内容抽取技术是 Web 新闻内容管理的重要应用基础; 在开放信息抽取领域 [7,8], 已有的基于 Web 结构特征的命名实体抽取等技术在开放环境下通常会失效, Web 新闻内容抽取技术可为开放命名实体抽取等任务提供 “干净的” 语料.

Web 新闻内容抽取是一个开放性问题. 做为大数据重要载体之一的 Web, 具有开放性、动态性、异构性等特点, 并且 Web 新闻出版缺少统一的出版标准, 导致 Web 新闻内容抽取任务成为一个开放性研究问题 [9]. 近年来, 随着 JavaScript, CSS, HTML5, AJAX 等 Web 技术的大量使用, 伴随着 Web 网页的结构、功能和语义的演化发展, 使 Web 内容抽取任务面临更大的挑战 [10].

自 20 世纪 90 年代中期以来, 相关研究人员已在 Web 信息抽取领域开展了大量的工作 [10~13]. 早期手工构造包装器技术通过专业技术人员针对特定的数据源编写抽取规则进行内容抽取, 其发挥作用的前提是数据源中的网页具有类似的组织结构, 该类技术费时费力, 对使用人员的专业要求极高. 另外, Web 网页的海量异构的特点, 使得这一假设前提已不再成立, 手工构造包装器技术在 Web 大数据时代已基本失效. 后来发展的包装器归纳技术通过监督学习、半监督学习和无监督学习等技术提高了构建包装器的自动化程度, 但该类技术被广为诟病之处在于学习的包装器难以应对网页模板的细微变化 [13]. 另外, 和手工构造包装器技术一样, 包装器归纳技术同样假设数据源中的网页具有类似的组织结构, 导致该类技术仅在限定条件下发挥作用, 难以胜任海量异构 Web 网页的抽取工作. 近年来发展的实时在线 Web 内容抽取方法能解决海量异构 Web 网页的抽取问题 [14], 其中: “实时” 意思是 “无需对网页进行预处理或预先知道它们的结构”, “在线” 的意思是 “能适应任何网页”. 但在包装器归纳技术能工作的场景, 因没有使用数据源中的网页具有类似的组织结构的假设前提, 该类技术在抽取精度上不占优势. 因此, 致力于提高该类技术的抽取精度是一项重要的研究工作.

在实时在线类 Web 内容抽取方法中, 抽取特征设计是研究的重点. 目前, 标签路径特征 [14] 和文本块密度特征 [15] 是实时在线类抽取方法中的两类性能优良的特征. 通过对网页解析树标签路径聚集的文本进行统计分析, 可以设计出在网页全局范围内对文本和噪音具有较强区分能力的抽取特征. 然而, 标签路径类特征存在设计上的不足, 该类特征难以区分内容块中的噪音和噪音块中的内容, 难以抽取内容块中低特征值内容, 且易将噪音块中的高特征值噪音误判为内容. 通过对网页解析树的子树聚集的文本进行统计分析, 可以设计出对内容块与噪音块具有较强区分能力的文本密度块特征. 然而, 基于该特征的抽取方法难以恰到好处地获得区分文本块和噪音块的阈值, 易将整块内容误判为噪音或整块噪音误判为内容, 导致该类特征在实际应用中存在鲁棒性不足的问题.

在一棵网页解析树中, 标签路径特征用于解析树中的文本节点, 通常将具有高标签路径特征值的文本节点判定为内容节点, 将低值文本节点判定为噪音节点; 文本块密度特征用于解析树中的文本块, 通常将具有高密度特征值的文本块判定为内容块, 将低值文本块判定为噪音块. 在抽取方法上, 基于标签路径特征的抽取方法抽取的是文本节点的内容, 而基于文本块密度特征的抽取方法抽取的是整个内容子树. 这两类特征及其相应的抽取方法存在着较大的差异, 难以直接融合. 本文要解决的问题是, 能否综合标签路径特征和文本块密度特征的优点设计一种性能更好的抽取特征及抽取方法?

针对该问题, 本文做了一些有益的探索研究工作, 贡献如下:

(1) 就我们目前已知的文献调研工作来看, 尚未见到用扩展标记有序树模型融合文本块密度特征

和标签路径特征. 本文引入扩展标记有序树表示模型, 根据每个文本节点均是某个文本块解析子树的叶节点的特性, 设计了适用于综合利用标签路径特征和文本块密度特征的 Web 内容抽取模型和算法框架 CEDP;

(2) 设计了一种有效的文本密度特征, 在此基础上, 设计了块密度加权标签路径特征, 以获得具有更强区分能力的综合特征;

(3) 基于块密度加权标签路径特征, 设计了在线 Web 新闻内容抽取算法 CEDP-NLTD, 对比分析了不同密度特征加权的效果, 并验证了该算法适用于海量、异构、多语言的 Web 新闻内容抽取工作.

本文第 2 节介绍了相关工作. 第 3 节设计了一种结合文本块密度和标签路径特征的 Web 内容抽取模型和算法框架. 第 4 节设计了一种文本密度特征, 以此基础上, 设计了块密度加权标签路径特征. 第 5 节给出了一种自动阈值计算方法. 以现实的数据源为实验数据, 第 6 节给出了抽取算法的性能评估和对比实验结果. 最后, 第 7 节对本文工作进行了总结和展望.

2 相关工作

互联网已经成为当今世界人们工作和生活形影不离的工具, 在未来很多年仍将继续渗透、改变和改善我们的生活和工作形态, 产生更多的产业和商业机遇. 然而, 随着互联网的迅猛发展, 互联网信息爆炸成为一个必须面对的问题. 如何帮助人们准确有效地从海量信息中找出所需要的信息, 进而高效地自动理解网络上出现的海量文本信息, 使得已有的信息处理和知识工程技术面临着严峻的考验, 成为 Web 大数据知识工程面临的挑战之一 [2,3,16].

目前 Web 上的信息大多是通过 HTML 形式进行展现, 缺乏对数据模式的描述, 使得应用程序难以直接解析并难以利用海量的 Web 信息. Web 信息抽取技术可有效解决该问题, Web 信息抽取是信息抽取方向的一个分支 [11], 将 Web 做为数据源, 从无结构化或半结构化的 Web 页面中抽取用户感兴趣的信息, 并输出具有清晰结构和语义的信息. Web 信息抽取是 Web 大数据知识工程的信息处理过程中的一个非常重要的步骤, 是开放数据和知识获取的重要基础.

Web 新闻抽取是一类面向领域面向特定属性的 Web 信息抽取, 根据应用需求的不同, 具体还可细分为新闻标题抽取 [17]、新闻时间抽取 [18]、新闻内容抽取 (包括标题、时间和正文, 但通常未做细致地识别和区分) [14].

有 3 类的抽取技术可用于 Web 新闻抽取: 手工构造包装器技术、包装器归纳技术、实时在线抽取技术. TSIMMIS [19], W4F [20] 和 XWRAP [21] 等手工构造包装器技术简单且直接, 能较好地解决特定领域的问题. 但手工构造费时费力, 需要专业人员维护抽取规则库, 自动化程度低, 构造代价高, 难以扩展和推广, 不适用于 Web 大数据环境下的开放信息抽取. 包装器归纳技术将监督学习技术 [13,22~25]、半监督学习技术 [26]、无监督学习技术 [27] 应用于包装器构造, 解决抽取规则的挖掘和学习问题, 可有效提高构造包装器的自动化程度. 然而, 包装器归纳技术需要假定被抽取的网页有特定的结构或共享某种结构. 其次, 面向新闻抽取这类任务, 需要针对成百上千的数据源各自构建不同的包装器, 工作量巨大. 另外, 网站模板的轻微变化均会导致已有的包装器失效, 检测与维护的这些包装器的代价极高. 尤为重要的是, 在 Web 大数据环境下, “被抽取的网页有特定的结构或共享某种结构” 这个假设不再成立, 导致包装器归纳技术难以有效地处理 Web 规模的信息抽取任务. 实时在线抽取技术, 如: CETR [10], CETD [15], CEPR [28], CEPF [14] 等抽取方法, 因无需对被抽取的网页对象有预先限定结构或内容的假设, 可较好地应用于 Web 大数据环境下的信息抽取任务, 近年来在 Web 信息抽取领域受到广泛关注.

CETR [10] 利用文本标签比特征进行 Web 内容抽取. 该方法将 HTML 网页视为一个纯文本文

件, 采用逐行的方式计算每行的文本标签比, 进而得到一个横坐标为行号、纵坐标为文本标签比特征值的直方图. 通过对结果直方图进行聚类, 获得内容类和噪音类. 因一维直方图不太适合使用聚类方法区分内容和噪音, CETR 将原始的方法扩展为二维模型, 以提高内容和噪音的区分度. 实验结果表明, CETR 优于 FE, KFE, BTE, DSC, ADSC, LQ, LP, CCB, MSS, VIPS 等抽取方法. CETR 直接对文本文件进行分析, 可以不受 HTML 解析器性能的限制 (在个别情况下, HTML 解析器不能成功地解析 HTML 文件). 缺点在于, 该方法仅使用了每行标签个数的结构特征, 没有充分地利用 HTML 网页的路径和子树等结构特征, 使得文本标签路径比特征难以成为一个区分内容和噪音的优良特征.

CETD [15] 设计了文本密度特征进行 Web 内容抽取. 该方法设计了两种密度特征和一种密度和特征去度量 HTML 网页解析树节点 (文本块) 的重要性. 一种密度特征是树节点 (文本块) 包含所有文本字符个数除以该节点为根的子树 (文本块) 中标签个数得到的文本密度特征; 另外一种密度特征是组合文本密度, 该特征在文本密度的基础上, 融合了链接文本、链接标签数等信息以加强区分内容和噪音的能力. 不同于 CETR 中使用的 Gauss 平滑和移动平均平滑, CETD 设计了一种基于密度和的平滑技术, 避免低密度文本块被判定为噪音. CETD 的优点在于, 该方法利用了网页解析树的子树结构特征, 比 CETR 的文本标签比特征能更好地区分内容和噪音. 另外, CETD 通过抽取高密度子树 (文本块), 保持了原有子树的结构, 可在原始树结构基础上原封不动地抽取内容. 然而, CETD 提出的启发式阈值计算方法, 易于误抽或漏抽子树 (整块的文本), 导致该方法的鲁棒性不足.

CEPR [28] 基于 Web 内容布局和它们的标签路径存在关联的现象, 设计了标签路径特征进行 Web 内容抽取. 该方法设计了文本标签路径比特征和扩展的文本标签路径比特征. 通过计算一个标签路径聚合的文本内容字符长度除以该标签路径在一个网页解析树中出现的次数得到文本标签路径比特征; 结合文本中标点符号等信息得到扩展的文本标签路径比特征, 以更好地区分内容和噪音. 为了抽取短文本内容, CEPR 设计了一种基于路径编辑距离加权的 Gauss 平滑方法, 在增强内链节点重要性的同时忽略新闻内容中的噪音节点. CEPR 设计的标签路径特征综合了标签路径在网页解析树全局范围内的统计信息, 使其具有良好的区分内容和噪音的能力. 然而, 该方法只从文本、标点符号和标签路径几个维度设计的特征仍存在抗干扰能力不足的缺点. 另外, 标签路径特征相对于解析树子树 (文本块) 的文本密度特征缺少区域范围内区分内容和噪音的优势.

为解决 CEPR 的抽取特征抗干扰能力不足的问题, CEPF [14] 设计了标签路径特征系, 可从不同视角区分网页内容和噪音. 为了有效融合不同标签路径特征, CEPF 将特征去冗问题转化为图的划分问题, 基于谱聚类实现了组合特征选择, 以去除冗余特征. 将选出的待融合特征相乘, 得到对内容和噪音区分能力更强的融合特征. 标签路径特征系的设计提高了标签路径特征在区分内容和噪音方面的抗干扰能力, 基于组特征选择的策略有效地提高了融合特征的质量. 实验结果表明, CEPF 的融合标签路径特征比 CEPR 的标签路径特征在抽取性能上有了较大的提高. 然而, 和 CEPR 具有类似的问题, 融合标签路径特征相对于解析树子树 (文本块) 的文本密度特征在区域范围内区分内容和噪音方面同样不占优势.

CETR, CETD, CEPR, CEPF 等实时在线抽取方法不依赖于任何特定的 HTML 标记, 不假定网页有任何特定的结构或共享特定的结构, 适用于开放环境下的 Web 新闻内容抽取任务. 已有研究结果表明, 结合从不同视角设计的特征有助于提升 Web 内容抽取的性能 [29]. 本文尝试研究结合目前两类在 Web 大数据环境下具有良好抽取性能的特征: 文本块密度特征和标签路径特征, 探索建立能有效结合这两类设计迥异的抽取特征的抽取模型, 设计适合 Web 大数据环境下的实时在线抽取方法, 并提高抽取性能.

3 结合文本块密度和标签路径特征的 Web 内容抽取模型

一个网页可以被解析成一棵树 (如 DOM 树), 为了便于描述, 本文以扩展标记有序树模型来表示这类解析树, 以此为基础, 定义文本块密度特征和标签路径特征等概念.

定义1 (扩展标记有序树) 设 L=$\{l_1, l_2, l_3, \dots\}$ 是一个有限字母表, 其中: l_i 为 HTML 标签. L 上的扩展标记有序树定义为一个九元式 T=(V, E, v_1, $\prec$, L, $l(\cdot)$, densityF$(\cdot)$, pathF$(\cdot)$, $c(\cdot)$), 其中: V 是树的节点集合, E 是树的边集合, $v_1 \in V$, $\prec$ 是树节点的兄弟关系集合, G=(V, E, v_1, $\prec$) 是一棵以 v_1 为根的有序树. 映射 l: $V \to L$ 是标记函数, $\forall v \in V$, $l(v)$ 为节点 v 上的标记. 映射 densityF: $V \to \mathbb{R}$ 是文本块密度特征函数, $\forall v \in V$, densityF(v) 为和节点 v 相联系的文本块密度特征值. 映射 pathF: $V \to \mathbb{R}$ 是标签路径特征函数, $\forall v \in V$, pathF(v) 为和节点 v 相联系的标签路径特征值. 节点 v 上的标记函数映射 c: $V \to$String 是内容函数, $\forall v \in V$, $c(v)$ 为节点 v 上的内容.

扩展标记有序树 T 上的映射 densityF$(\cdot)$, pathF$(\cdot)$, $c(\cdot)$ 可根据抽取任务的需要具体定义.

定义2 (文本块、块文本子树) 设 T 是 L 上的扩展标记有序树, T_{sub} 是 T 上以 v_{sub} 为根的子树, $V_{T_{\text{sub}}}$ 是子树 T_{sub} 上所有节点的集合. 令 t=$+_{v \in V_{T_{\text{sub}}} - \{v_{\text{sub}}\}} c(v)$, 其中: "+" 为字符串连接运算. 若 t 不为空串, 则称子树 T_{sub} 为文本块或块文本子树.

文本块、块文本子树的定义表明, T_{sub} 是一棵以 v_{sub} 为根的子树, T_{sub} 是一个文本块或块文本子树当且仅当 v_{sub} 的某一棵孩子子树中存在非空文本节点.

设 T 为一棵扩展标记有序树, V_T 为 T 上所有节点的集合, 对 $\forall v \in V_T$, 设 T_{sub} 是一棵以 v 为根的子树, 可根据 T_{sub} 上的内容信息和结构信息设计块密度特征函数并计算 densityF(v). 通常情况下, 若 T_{sub} 是一个块文本子树, 则 densityF(v) 的值大于 0, 反之为 0.

定义3 (标签路径) 设 T 是 L 上的以 v_1 为根的扩展标记有序树. $\forall v \in V$, 则在树 T 上存在唯一从 v_1 到达 v_k 的节点序列 $\langle v_1, v_2, \dots, v_k \rangle$, 其中 parent$(v_i)$=$v_{i-1}$, $2 \leqslant i \leqslant k$, v_k=v. 称 $l(v_1) \cdot l(v_2) \cdot \ldots \cdot l(v_k)$ 为节点 v 的标签路径, 记为 path(v).

由标签路径的定义可知, 一棵扩展标记有序树上存在有限可数个不同标签路径, 每一条标签路径从根节点出发, 逐层根据标签进行匹配, 最终可聚集一个匹配的树节点集合.

设 T 为一棵扩展标记有序树, P_T 为 T 上所有标签路径的集合. 对 T 上的任意节点 v, 可知 path(v) 是集合 P_T 的一个元素. 设 $V_{\text{path}(v)}$ 是标签路径 path(v) 在树 T 上聚集的节点集合, 可根据节点 v 上的内容信息、集合 $V_{\text{path}(v)}$ 上的内容信息、path(v) 在 T 上的结构和数量信息定义标签路径特征函数并计算 pathF(v). 通常情况下, 若 $V_{\text{path}(v)}$ 不为空集且其中存在内容不为空串的节点, 则 pathF(v) 的值大于 0, 反之为 0.

对扩展标记有序树上的任意节点 v, 我们可以计算节点 v 的块密度特征值 densityF(v) 和标签路径特征值 pathF(v), 通过设计一个特征融合函数, 可将这两个特征值融合成一个特征值. 若块密度特征值、标签路径特征值和特征融合函数设计得足够好, 则该特征可有效区分内容和噪音. 不妨设内容节点倾向于高特征值、噪音节点倾向于低特征值, 则可通过一个区分阈值函数在该特征上进行判定. 若特征值高于阈值则将该节点判断为内容节点, 反之则判定为噪音节点.

定义4 ((节点, 内容) – 规范文本节点序列) 设树 T 的先序遍历节点序列为 $\langle v_1, v_2, \dots, v_n \rangle$, 若 $\langle v_{i1}, v_{i2}, \dots, v_{im} \rangle$ 为 $\langle v_1, v_2, \dots, v_n \rangle$ 的子序列. 对子序列中任意节点 v_{ij} $(1 \leqslant j \leqslant m)$, 有 $c(v_{ij}) \neq$"", 对 $\forall v \in V_T - \bigcup_{ij}\{v_{ij}\}$, 有 $c(v)$="". 则称 nts=$\langle (v_{i1}, c(v_{i1})), (v_{i2}, c(v_{i2})), \dots, (v_{im}, c(v_{im})) \rangle$ 为树 T 的 (节点, 内容) – 规范文本节点序列, 称 nts 的子序列为 (节点, 内容) – 规范文本节点子序列.

结合文本块密度和标签路径特征的 Web 内容抽取模型: 给定网页 wp, 块密度特征函数 densityF(·), 标签路径特征函数 pathF(·), 特征融合函数 combine(·,·,·), 区分阈值函数 thresh(·). 网页 wp 的扩展标记有序树为 T_{wp}, 节点序列 nts=$\langle(v_1,\ c(v_1)),\ (v_2,\ c(v_2)),\ \ldots,\ (v_n,\ c(v_n))\rangle$ 是 T_{wp} 上的 (节点, 内容) – 规范文本节点序列, 节点序列 ies=$\langle(v_{i1},\ c(v_{i1})),\ (v_{i2},\ c(v_{i2})),\ \ldots,\ (v_{im},\ c(v_{im}))\rangle$ 是 T_{wp} 上的一个 (节点, 内容) – 规范文本节点子序列. 记 nts 中节点的集合为 V_{nts}, ies 中节点的集合为 V_{ies}, V_{nts} 和 V_{ies} 中的元素满足如下条件:

(1) $\forall v \in V_{\text{ies}}$, 有 $\text{combine}(v, \text{densityF}(\cdot), \text{pathF}(\cdot)) \geqslant \text{thresh}(T_{\text{wp}})$;

(2) $\forall v \in V_{\text{nts}} - V_{\text{ies}}$, 有 $\text{combine}(v, \text{densityF}(\cdot), \text{pathF}(\cdot)) < \text{thresh}(T_{\text{wp}})$,

则称 ies 为网页 wp 结合文本块密度和标签路径特征抽取得到的 (节点, 内容) – 规范文本节点子序列, 序列 ts=$\langle c(v_{i1}), c(v_{i2}), \ldots, c(v_{im})\rangle$ 为抽取得到的内容.

根据抽取模型, 算法 1 给出了结合文本块密度和标签路径特征的 Web 内容抽取算法框架 CEDP. 该算法首先遍历网页 wp 的解析树获取 (节点, 内容) – 规范文本节点序列. 对序列中每个 $(v_i, c(v_i))$ 对, 计算 v_i 的块密度特征值 densityF(v_i) 和标签路径特征值 pathF(v_i), 根据 combine 函数计算融合特征值 $f(v_i)$, 使用函数 smoothing 对融合特征值进行平滑, 如平滑后的融合特征值 $sf(v_i)$ 大于等于阈值, 则判定 v_i 为内容节点, 并将 $c(v_i)$ 作为内容抽取, 反之则判定 v_i 为噪音节点, 并忽略 $c(v_i)$. 这里的阈值通过 thresh 函数计算 T_{wp} 上所有文本节点的相关统计特征而得到. 设 T_{wp} 有 m 个节点, 其中包括 n 个文本节点. 计算块密度特征、标签路径特征和融合特征在遍历树过程中完成, $O(f_{\text{densityF}}(m))$, $O(f_{\text{pathF}}(m))$, $O(f_{\text{combine}}(m))$ 分别为相应的最坏时间复杂度. thresh 函数处理数据的规模为 n, 最坏时间复杂度记为 $O(f_{\text{thresh}}(n))$. 算法 CEDP 的最坏时间复杂度为 $O(n \times (f_{\text{densityF}}(m) + f_{\text{pathF}}(m) + f_{\text{combine}}(m) + f_{\text{thresh}}(n)))$. CEDP 抽取模型和算法框架简单、易于实现, 内容抽取质量取决于融合特征区分内容噪音的性能和合理的阈值, 因此, 如何设计具有良好性能的块密度特征、标签路径特征、融合函数、阈值函数成为 CEDP 模型和算法的关键问题.

算法 1 CEDP

Input: 网页 wp, densityF(·), pathF(·), combine(·,·,·), thresh(·), smoothing(·);
Output: 文本内容 content;

```
1: 解析 wp 得到解析树 T_wp, content ← "";
2: nts ← ⟨(v_1, c(v_1)), (v_2, c(v_2)), ..., (v_n, c(v_n))⟩ //计算 T_wp 的 (节点, 内容) – 规范文本节点序列 nts;
3: for i = 1 to n do
4:     f(v_i) ← combine(v_i, densityF(·), pathF(·));
5: nfts ← ⟨(f(v_1), c(v_1)), (f(v_2), c(v_2)), ..., (f(v_n), c(v_n))⟩;
6: s_nfts ← smoothing(nfts), 并记 s_nfts 为 ⟨(sf(v_1), c(v_1)), (sf(v_2), c(v_2)), ..., (sf(v_n), c(v_n))⟩;
7: for i = 1 to n do
8:     if sf(v_i) ⩾ thresh(T_wp) then
9:         content ← content + c(v_i);
10: output content.
```

4 块密度加权标签路径特征设计

设 T 是 L 上的扩展标记有序树, V 是 T 上所有节点的集合. T_{sub} 是 T 上以 v_{sub} 为根的子树, T_{sub} 上所有节点的集合为 V_{sub}. 设 v 为树 T 上的节点, 节点 v 上的文本为 $c(v)$、文本长度为 length($c(v)$)、标点符号个数为 PNum($c(v)$). 标签路径 p 在树 T 上的聚集的节点集合为 accNodes(p)、层次数为

level(p) (可量化标签路径 p 的修饰程度). 本节将给出块密度特征函数、标签路径特征函数、融合函数的设计, 从而得到一种新颖的块密度加权标签路径特征.

4.1 块密度特征函数

块字符数 CNumBlk(v_{sub}) 定义为块子树中除去根节点以外的所有节点包含文本的长度之和:

$$\text{CNumBlk}(v_{\text{sub}}) = \sum_{v \in V_{\text{sub}} - \{v_{\text{sub}}\}} \text{length}(c(v)).$$

块标签数 TagNumBlk(v_{sub}) 定义为块子树中除去根节点以外的所有节点标签数之和:

$$\text{TagNumBlk}(v_{\text{sub}}) = |V_{\text{sub}} - \{v_{\text{sub}}\}|.$$

在块字符数和块标签数的定义中没有将子树根节点统计在内, 是为了避免将单独的长文本叶节点判定为文本块.

令 $V_{\text{sub}}^{\text{link}} = \{v | \forall v \in V_{\text{sub}},\ l(v)$是一个链接标签, 且 v 所有祖先节点上的标记不为链接标签$\}$, 块链接字符数定义为以 $V_{\text{sub}}^{\text{link}}$ 中节点为根的链接块子树包含的文本长度之和:

$$\text{LCNumBlk}(v_{\text{sub}}) = \sum_{v \in V_{\text{sub}}^{\text{link}}} (\text{CNumBlk}(v) - \text{length}(c(v))).$$

基于块字符数、块链接字符数、块标签数定义一种块密度特征 —— 非链接文本密度:

$$\text{densityF}(v) = \begin{cases} \text{CNumBlk}(v) - \text{LCNumBlk}(v), & \text{TagNumBlk}(v) = 0, \\ (\text{CNumBlk}(v) - \text{LCNumBlk}(v))/\text{TagNumBlk}(v), & \text{TagNumBlk}(v) \neq 0. \end{cases}$$

非链接文本密度对网页解析树中每个节点包含的非链接文本密度进行度量. 通常赋予高值给块子树中包含长文本、简单格式化、有较少链接的节点, 赋予低值给块子树中包含短文本、较多格式化、有较多链接的节点. 显然, 内容节点 (块子树) 通常具有较高的非链接文本密度值, 而噪音节点 (块子树) 倾向获得较低的非链接文本密度值. 非链接文本密度综合了文本、格式化、链接 3 个维度的信息, 具有较好的区分内容和噪音的能力.

4.2 标签路径特征函数

在一个 Web 新闻中, 内容通常具有相似的标签路径, 噪音也具有相似的路径; 内容通常包括较长的文本、较多的标点符号、较少的修饰; 噪音通常包括较短的文本、较少的标点符号、较多的修饰. 根据这些分析, 文献 [28] 设计了标签路径特征系和基于谱聚类的融合特征.

文本标签路径长度特征 TPL(p) 定义为 p 聚集的节点的文本长度之和:

$$\text{TPL}(p) = \sum_{v \in \text{accNodes}(p)} \text{length}(c(v)).$$

文本标签路径比特征 TPR(p) 为 p 聚集的节点的文本长度之和与标签路径数的比值:

$$\text{TPR}(p) = \frac{\sum_{v \in \text{accNodes}(p)} \text{length}(c(v))}{|\text{accNodes}(p)|}.$$

文本标签路径层次比特征 TPLR(p) 是 p 聚集的节点的文本长度之和与标签路径层次数的比值:

$$\mathrm{TPLR}(p)=\frac{\sum_{v\in\mathrm{accNodes}(p)}\mathrm{length}(c(v))}{\mathrm{level}(p)}.$$

标点标签路径长度特征 PPL(p) 定义为 p 聚集的节点的标点符号个数之和:

$$\mathrm{PPL}(p)=\sum_{v\in\mathrm{accNodes}(p)}\mathrm{PNum}(c(v)).$$

标点标签路径比特征 PPR(p) 为 p 聚集的节点的标点符号个数之和与标签路径数的比值:

$$\mathrm{PPR}(p)=\frac{\sum_{v\in\mathrm{accNodes}(p)}\mathrm{PNum}(c(v))}{|\mathrm{accNodes}(p)|}.$$

标点标签路径层次比 PPLR(p) 是 p 聚集的节点的标点符号个数之和与标签路径层次数的比值:

$$\mathrm{PPLR}(p)=\frac{\sum_{v\in\mathrm{accNodes}(p)}\mathrm{PNum}(c(v))}{\mathrm{level}(p)}.$$

在抽取不同 Web 新闻网页时, 标签路径特征系 {TPL, TPR, TPLR, PPL, PPR, PPLR} 中不同标签路径特征有其独特的优势. 另外, 这些标签路径特征之间也有相似性, 去除一些冗余特征可提高抽取的效率和精度. 将每个标签路径特征视为 n 维向量空间中的一个点, 采用 Gauss 函数度量两点 (特征) 间的相似度, 得到一个无向加权图. 选取谱聚类算法解决图的划分问题, 可得到聚类结果 $C=\{C_1,C_2,\ldots,C_k\}$ $(1\leqslant k\leqslant m)$, m 为标签路径特征系的特征数量, k 值由谱聚类算法确定. 同类标签路径特征间存在冗余, 不同类标签路径特征间没有冗余. 从每个类别任选一标签路径特征构成特征子集 $\{\mathrm{PF}_1,\mathrm{PF}_2,\ldots,\mathrm{PF}_k\}$, 其中: PF_i $(1\leqslant i\leqslant k)$ 是标签路径特征系中互不相同的特征. 采用乘法将这 k 个特征和 σ_{cs}, σ_{ps} 融合为一个具有较强区分内容和噪音能力的融合特征, 其中: σ_{cs} 为标签路径聚合的文本节点内容长度标准差, σ_{ps} 为标签路径聚合的文本节点标点符号数量标准差. 由此, 得到融合特征为

$$\mathrm{FPF}(p)=\prod_{1\leqslant i\leqslant k}\mathrm{PF}_i(p)\times\sigma_{\mathrm{cs}}\times\sigma_{\mathrm{ps}}.$$

根据标签路径特征系和基于谱聚类的融合特征, 设计标签路径特征函数如下:

$$\mathrm{pathF}(v)=\mathrm{FPF}(\mathrm{path}(v)).$$

通过遍历网页的解析树, 可以获得所有的标签路径, 并统计每个文本节点的文本长度、包含的标点符号的个数. 因此, 每个标签路径的所有标签路径特征值和融合特征值都可以方便地计算出来.

4.3 融合函数

在块密度函数 densityF($\cdot$) 和标签路径函数 pathF($\cdot$) 的基础上, 设计 combine($\cdot,\cdot,\cdot$) 函数如下:

$$\mathrm{combine}(v,\mathrm{densityF}(\cdot),\mathrm{pathF}(\cdot))=\frac{\mathrm{Max}(\{\mathrm{densityF}(v')|v'\text{ is }v\text{ or an ancestor of }v\})}{\mathrm{Max}(\{\mathrm{densityF}(v')|v'\in V\})}\times\mathrm{pathF}(v).$$

该函数由两部分相乘, 第一项为从 v 到树根路径上所有节点的最大块密度函数值与树上所有节点的最大块密度函数值的比值, 值的范围在 [0, 1] 之间, 第二项为 pathF(v). 该融合函数的设计可以看作

是一种基于块密度特征对标签路径特征的加权, 融合块密度特征的优点以改善标签路径特征区分内容和噪音的能力. 因此, 称这种经融合函数得到的特征为块密度加权标签路径特征.

标签路径特征和块密度特征本质上都是基于评分策略设计的特征. 标签路径特征通过评分策略使得聚合内容的标签路径特征具有相对较高的特征值, 聚合噪音的标签路径特征具有相对较低的特征值. 因内容块和噪音块中的同一标签路径具有相同的特征值, 从而导致误抽取和漏抽取现象的发生. 具有最大密度的文本块通常一定包含内容块, 基于块密度特征的加权策略考虑了标签路径是内容块中的标签路径的可能性, 通过当前路径所在文本块的最大块密度特征除以全局最大块密度特征的值对标签路径值加权, 修正仅通过标签路径特征值进行评分的不足, 从而有效地结合了块密度特征和标签路径特征进行评分.

5 自动阈值计算方法

设 p 为树 T 的标签路径, 其基于谱聚类的融合特征值为 $\mathrm{FPF}(p)$. 树 T 有 L 个不同的包含文本节点的标签路径, 按 FPF 特征值非递减排序得到有序集合 P_T, 第 i 条标签路径记为 P_T^i $(1 \leqslant i \leqslant L)$, 总特征值为 $\sum_{1\leqslant i\leqslant L}\mathrm{FPF}(P_T^i)$. 以下使用归一化的融合特征值直方图, 并视其为文本节点的标签路径特征值 (表征内容) 的概率分布:

$$\mathrm{pr}_i = \frac{\mathrm{FPF}(P_T^i)}{\sum_{1\leqslant i\leqslant L}\mathrm{FPF}(P_T^i)}, \quad \mathrm{pr}_i \geqslant 0, \quad \sum_{1\leqslant i\leqslant L}\mathrm{pr}_i = 1.$$

因聚集内容的标签路径通常具有较高的特征值, 聚集噪音的标签路径具有较低的特征值. 设候选阈值范围为 $[\alpha\delta\sigma_T, \beta\delta\sigma_T]$, 我们试图在阈值范围内搜索一个最佳阈值 $\gamma^*\delta\sigma_T(\alpha \leqslant \gamma \leqslant \beta)$, 其中 $\alpha, \beta, \gamma \in \mathbb{N}, \delta \in \mathbb{R}$, σ_T 为 T 上所有文本节点块密度加权标签路径特征值的标准差. 设候选阈值 $\gamma\delta\sigma_T$ 从有序集合 P_T 的第 k_γ 条标签路径处将其划分为两类: C_0 和 C_1, 当 $i \leqslant k_\gamma$ 时, $\mathrm{FPF}(P_T^i) < \gamma\delta\sigma_T$; 反之, 当 $i > k_\gamma$ 时, $\mathrm{FPF}(P_T^i) \geqslant \gamma\delta\sigma_T$, $i \in [1, L]$. C_0 类表示聚集噪音的标签路径类, C_1 表示聚集内容的标签路径类.

C_0 类的出现概率为

$$\omega_0(k_\gamma) = \Pr(C_0) = \sum_{i=1}^{k_\gamma}\mathrm{pr}_i.$$

C_1 类的出现概率为

$$\omega_1(k_\gamma) = \Pr(C_1) = \sum_{i=k_\gamma+1}^{L}\mathrm{pr}_i.$$

C_0 类的平均特征值为

$$\mu_0(k_\gamma) = \sum_{i=1}^{k_\gamma}\mathrm{FPF}(P_T^i)\times\Pr(P_T^i|C_0) = \sum_{i=1}^{k_\gamma}\mathrm{FPF}(P_T^i)\times\mathrm{pr}_i/w_0(k_\gamma).$$

C_1 类的平均特征值为

$$\mu_1(k_\gamma) = \sum_{i=k_\gamma+1}^{L}\mathrm{FPF}(P_T^i)\times\Pr(P_T^i|C_1) = \sum_{i=k_\gamma+1}^{L}\mathrm{FPF}(P_T^i)\times\mathrm{pr}_i/w_1(k_\gamma).$$

集合 P_T 的累积概率和平均特征值为

$$\mu_\tau = \sum_{i=1}^{L} \mathrm{FPF}(P_T^i) \times \mathrm{pr}_i.$$

设计目标函数为 C_0 和 C_1 这两类的类间方差:

$$\sigma_B^2(\gamma) = \omega_0(k_\gamma)(\mu_0(k_\gamma) - \mu_\tau)^2 + \omega_1(k_\gamma)(\mu_1(k_\gamma) - \mu_\tau)^2.$$

假设一个好的阈值将会把标签路径集合分为两类, 聚集噪音节点的 C_0 类和聚集内容节点的 C_1 类, 计算阈值的问题则转化为一个最大化类间方差的优化问题 [30], 即寻找最优的 γ^* 使得目标函数 $\sigma_B^2(\gamma)$ 取最大值. γ^* 对应的 $\gamma^*\delta\sigma_T$ 即为所求阈值. 因此, 设计阈值函数 $\mathrm{thresh}(T)$ 如下:

$$\mathrm{thresh}(T) = \gamma^*\delta\sigma_T, \quad \text{where } \gamma^* = \underset{\alpha\leqslant\gamma\leqslant\beta}{\mathrm{ArgMax}}(\sigma_B^2(\gamma)); \quad \alpha,\ \beta \text{ and } \delta \text{ are constants.}$$

6 抽取性能评估

本文评估的算法均为实时在线算法, 不需要训练语料事先学习一个模型. 因此, 所有标注好的新闻网页语料可全部做为测试集. 实验的硬件配置为处理器: Intel(R) Core(TM) i7-2640M CPU @ 2.80 GHz, 2.80 GHz; 内存: 4GB RAM; 操作系统: Windows 7 Professional x64; 开发平台: JDK 1.8.0_91. $\mathrm{thresh}(T)$ 函数中的 $\alpha = 5, \beta = 15, \delta = 0.1$, 即在 $[0.5\sigma_T, 1.5\sigma_T]$ 范围搜索最佳阈值, 超参数 α, β, δ 的设置是在相关研究工作 [10, 14] 成功尝试的基础上人为经验的设置. smoothing 函数采用 CEPR 和 CEPF 使用的基于标签路径编辑距离的加权 Gauss 平滑方法. 我们开源了 CEDP 系统, 并共享了实验数据集[1)].

6.1 数据集与评价标准

本文实验使用的数据集来自 CEPF. 包括: 2 个竞赛用数据集, 13 个新闻网站数据集, 3 个微博数据集, 共 6253 个网页, 具有广泛的代表性. 以下分 3 类进行介绍.

(1) CleanEval 数据集: 该数据集是 CleanEval 竞赛 [31] 使用的数据, 共 1903 个网页. 包括一个包含 932 个网页的英文数据集 CleanEval-EN 和一个包含 971 个网页的中文数据集 CleanEval-ZH. CleanEval 数据集的网页分布范围和类型极广, 存在一些难以处理的极端情况, 也存在标注不准确的问题, 对目前大多数 Web 信息抽取算法都形成挑战. 这个数据集是很多 Web 信息抽取算法进行性能评估的基准数据集之一. 例如, CETR 选取了 CleanEval 数据集部分网页 (741 个英文网页和 713 个中文网页), CEPR 使用了 CETR 同样的数据集做测试, CETD 选取了 CleanEval 数据集中的英文数据集做测试.

(2) 新闻网站数据集: 该数据集包括 13 个新闻网站的 3450 个新闻网页. 具体分布如下: NY Post (300), Suntimes (300), Techweb (250), Tribune (300), Nytimes (150), Freep (50), BBC (300), Reuters (300), Yahoo! (300), Sina (300), People (300), 163 (300), Xinhua (300). 其中: 数据集 NY Post, Suntimes, Techweb, Tribune, Nytimes, Freep, BBC, Reuters 也是 CETR 和 CEPR 所使用的数据集, 但它们在每个网站仅选用了 50 个网页. 为了更好地验证算法的有效性, CEPF 在 CETR 和 CEPR 原有数据集基础上做了不同程度的扩展, 并增加了英文数据集 Yahoo! 和中文数据集人民网 (People)、网易 (163)、新华网 (Xinhua).

1) https://github.com/hfut-dmic/CEDP.

改为了脚注，请核实。对的，已确认。

(3) 微博数据集: 在一个应用领域上设计的信息抽取技术很难复用到其他的应用领域 [12]. 为了测试在 Web 新闻抽取领域设计的算法 CEPF 在处理新型 Web 应用领域的可复用性, CEPF 选取 3 个微博网站共 900 个微博网页进行了测试分析, 具体分布为: 腾讯微博 (TunxunWb, 300)、新浪微博 (SinaWb, 300) 和搜狐微博 (SohuWb, 300).

和 CETR, CETD, CEPR, CEPF 一样, 本文采用精度 P (precision)、召回率 R (recall) 和 F 值 F_score 做为 Web 新闻内容抽取的性能评估指标. 抽取的结果、手工标记的结果都看作是字符串/字符 (英文是空格分隔的符号串, 中文是汉字和字母) 的集合.

$$P = \frac{|S_e \bigcap S_l|}{|S_e|}, \quad R = \frac{|S_e \bigcap S_l|}{|S_l|}, \quad F = \frac{2 \times P \times R}{P + R},$$

其中, S_e 为抽取结果集合, S_l 为手工标记结果集合, $S_e \bigcap S_l$ 为抽取结果中应该被抽取的内容集合. 精度 P 定义为抽取结果中被正确抽取部分的比例, 召回率 R 定义为应该抽取的内容被正确抽取的比例, F 值是一个综合评价指标.

6.2 对比实验

本文在 Java 环境下实现了 CEDP 模型和方法, 网页解析器使用的是 HTML Parser. 记本文提出的基于非链接文本块密度加权标签路径特征抽取方法为 CEDP-NLTD. 选取了 CETR, CEPR, CETD, CEPF 这几个目前主要的在线实时抽取算法为对比对象, 并均采用相应算法的原作者提供的源代码. CETR 方法使用的是原文给定的经验阈值参数. CEPR 用的是 CEPF 和其做对比的自动阈值版本, 记为 CEPR-AT. CETD 是一种自动阈值方法, 无需提供提供经验参数. 其中 CETD-QT 使用的是作者提供的源代码, 该版本采用 QT C++ 实现, 由于该实现使用的 HTML 解析器对一些网页不能正常解析, 并存在未处理字符编码自动识别等一些细节问题, 导致该实现在部分数据集上无法获得结果. 因此, 本文对照作者提供的源码在 Java 环境下实现了 CETD-Jsoup 版本, 使用 Jsoup 解析 HTML 网页 (Jsoup 解析器性能良好且能提供 CETD 方法所必需的节点删除操作), 并完善了网页字符编码自动识别等细节问题. CEPF 是一种自动阈值方法, 无需提供经验参数, 在 Java 环境下实现, 网页解析器使用的是 HTML Parser.

另外, 我们也给出了其他块密度定义的密度特征函数 densityF(·), 分别是 CETD 方法设计的 3 种块密度: TD (text density), CTD (composite text density) 和 DSum (density sum), 根据这 3 种块密度特征加权的 CEDP 算法分别记为 CEDP-TD, CEDP-CTD 和 CEDP-DSum. 本文探讨不同密度特征对 CEDP 模型抽取效果影响作用的同时, 也展现了 CEDP 模型和方法良好的可扩展性.

表 1~3 分别给出了每个算法在各个数据集上抽取性能的 F 值、精度和召回率. 表的每行是数据集, 最后一行是平均性能. 表的每列是各个算法.

从表 1 中可以看出, 本文所提的基于非链接文本块密度加权标签路径特征的 CEDP-NLTD 算法具有最好的平均性能, 优于所有的对比对象. 基于文本块密度加权标签路径特征的 CEDP-TD 算法的抽取性能排第 2 位, 对比对象之一 CEPF 的抽取性能排第 3 位. 仔细观察表 1 可以发现, 在 18 个数据集上, CEDP 系列的 4 个算法 CEDP-TD, CEDP-CTD, CEDP-DSum, CEDP-NLTD 在 10 个数据集上取得最好的性能, 说明结合块密度和标签路径特征的 CEDP 模型和方法在抽取性能方面具有比较普遍的优势. 然而, CEDP-CTD 和 CEDP-DSum 在 CleanEval 数据集上的表现不佳, 拖累了这两个算法在整体数据集上的表现. 仔细分析发现, 为了加大内容块和噪音块的区分值, CTD 和 DSum 块密度特征值之间的差异较大, CEDP 是一种基于标签路径特征的加权模型, 大幅度的加权操作会破坏标签

表 1　每个算法在各个数据集上的抽取 F 值 (优胜者标记为粗体)

Table 1　F_score for each algorithm in each source (the best sources are marked in bold)

DataSet	CETR	CEPR-AT	CETD-QT	CETD-Jsoup	CEPF	CEDP-TD	CEDP-CTD	CEDP-DSum	CEDP-NLTD
CleanEval-En	88.30%	75.33%	83.89%	**91.49%**	88.39%	88.38%	66.10%	66.78%	88.24%
CleanEval-Zh	83.36%	75.65%	N/A	**87.69%**	86.86%	87.04%	74.13%	74.07%	86.94%
NY Post	58.19%	81.02%	82.78%	83.97%	90.04%	89.00%	89.29%	**90.19%**	89.36%
Freep	70.36%	86.00%	74.41%	74.65%	87.79%	88.97%	90.18%	**90.70%**	90.16%
Suntimes	82.20%	85.90%	90.37%	90.07%	94.08%	94.41%	95.03%	**95.12%**	94.58%
Techweb	74.56%	88.86%	77.86%	77.35%	90.70%	91.00%	89.53%	90.70%	**91.08%**
Tribune	89.83%	90.32%	N/A	**95.47%**	95.21%	94.86%	95.08%	94.90%	94.93%
Nytimes	91.14%	86.91%	**96.26%**	96.25%	92.31%	92.07%	91.09%	90.50%	92.16%
BBC	72.76%	80.13%	89.45%	84.85%	89.53%	90.26%	90.56%	**90.87%**	90.65%
Reuters	71.73%	84.26%	N/A	77.67%	**94.40%**	94.24%	78.09%	79.22%	94.35%
Yahoo!	82.06%	84.96%	89.85%	85.88%	89.33%	90.91%	90.42%	**91.75%**	91.10%
Sina	73.99%	90.63%	N/A	89.44%	96.92%	97.36%	97.33%	**97.45%**	97.34%
People	86.23%	85.32%	82.40%	82.14%	89.27%	95.08%	95.10%	94.84%	**95.22%**
163	38.28%	**88.56%**	N/A	53.15%	79.84%	80.44%	80.63%	80.16%	80.57%
Xinhua	83.32%	81.24%	91.18%	90.57%	**95.08%**	94.84%	94.82%	94.92%	94.82%
TunxunWb	79.36%	17.75%	86.28%	86.90%	83.72%	**87.99%**	86.34%	86.71%	87.17%
SinaWb	57.99%	18.88%	N/A	77.01%	79.38%	85.56%	82.57%	79.90%	**85.88%**
SohuWb	87.16%	92.32%	N/A	87.57%	**93.51%**	92.22%	92.28%	92.38%	92.27%
Average	76.16%	77.45%	N/A	84.01%	89.80%	90.81%	87.70%	87.84%	**90.93%**

路径特征原有的鲁棒性特点. 相比较而言, TD 和 NLTD 特征值之间的差异不是太大, 这两类块密度函数在原有标签路径特征基础上进行幅度不大的加权易于取得比较好的效果. CEDP-TD 仅在 1 个数据集上取得一个最优性能, 但平均性能却排在第 2, 也表明在 CEDP 模型中, 密度特征函数值之间的差异不大时能提升抽取性能. CEDP-NLTD 性能优于 CEDP-TD 表明综合了链接和文本信息的密度特征优于仅考虑文本信息的密度特征.

从表 1 中还可以发现, CETD 的两个版本 CETD-QT 和 CETD-Java 分别在 4 个数据集上取得最优的抽取性能, 但平均性能不理想. CETD-Jsoup 比 CEPF 的 F 值差 5.79%, 比 CEDP-NLTD 的 F 值差 6.92%. 表明基于密度特征的抽取方法在鲁棒性方面存在不足. CETD-QT 和 CETD-Java 是同一种方法的不同实现, 但抽取性能仍有差别, 经过细致分析发现, 对同一网页不同解析器解析得到的解析树不完全相同, 从而导致这类基于树结构分析的抽取方法的抽取结果不同.

-Jsoup?与表中项目不一致，
作者核。对，改为CETD-Jsoup

从表 1 和 2 中发现, 虽然 CEDP-NLTD 仅在 1 个数据集上取得最好的抽取精度, 一直维持在较高精度水平. CEPR-AT 虽在 13 个数据集上取得最好的精度, 但平均 F 多种标签路径特征的 CEPF 以及 CEDP-TD 在大多数数据集上也一直能维持较高的 较高的 F 值.

从表 1 和 3 中发现, CEPF, CEDP-CTD 和 CEDP-NLTD 在大多数数据集上维持较高的召回率并能取得较高的 F 值. CETD-QT 和 CETD-Jsoup 在 15 个数据集上取得最高的召回率, 但 F 值性能不好, 这说明了 CETD 算法的启发式阈值算法过偏于低值.

在时间性能方面, 设网页的解析树有 m 个节点, 其中包括 n 个文本节点 ($n < m$). 对本文计

表 2 每个算法在各个数据集上的抽取精度 (优胜者标记为粗体)

Table 2 Precision for each algorithm in each source (the best sources are marked in bold)

DataSet	CETR	CEPR-AT	CETD-QT	CETD-Jsoup	CEPF	CEDP-TD	CEDP-CTD	CEDP-DSum	CEDP-NLTD
CleanEval-En	89.42%	**95.96%**	75.16%	89.86%	92.93%	93.39%	70.80%	70.07%	93.70%
CleanEval-Zh	79.60%	85.23%	N/A	85.18%	88.37%	88.82%	80.32%	79.89%	**88.90%**
NY Post	42.68%	**98.28%**	72.91%	74.76%	92.10%	90.80%	91.38%	93.24%	91.42%
Freep	57.38%	**86.75%**	59.52%	59.92%	80.66%	83.07%	85.67%	85.65%	85.26%
Suntimes	70.82%	**98.10%**	83.01%	82.64%	94.73%	95.28%	96.04%	96.24%	95.44%
Techweb	60.29%	**94.04%**	63.88%	63.36%	86.96%	87.22%	87.38%	87.22%	87.39%
Tribune	84.99%	**99.43%**	N/A	92.49%	96.32%	96.26%	96.64%	96.34%	96.31%
Nytimes	88.98%	**99.73%**	96.53%	95.86%	96.13%	96.85%	96.93%	96.37%	96.91%
BBC	60.11%	**97.83%**	81.93%	74.57%	91.75%	93.60%	94.80%	95.07%	94.60%
Reuters	58.48%	**98.13%**	N/A	63.65%	95.41%	95.05%	79.30%	81.04%	95.27%
Yahoo!	72.67%	**97.83%**	83.63%	76.62%	93.04%	93.29%	93.40%	94.70%	93.41%
Sina	59.09%	**98.57%**	N/A	81.93%	96.29%	97.64%	97.72%	97.74%	97.65%
People	77.04%	95.11%	70.21%	69.74%	84.47%	96.14%	**97.13%**	95.02%	96.65%
163	24.40%	**99.29%**	N/A	37.72%	73.59%	74.47%	74.90%	74.45%	74.66%
Xinhua	72.10%	94.72%	84.39%	83.28%	96.26%	96.51%	**96.60%**	96.49%	96.59%
TunxunWb	66.37%	94.25%	88.89%	88.40%	**99.36%**	98.87%	98.87%	98.87%	98.87%
SinaWb	41.81%	**86.84%**	N/A	64.43%	66.90%	77.86%	85.34%	67.65%	80.74%
SohuWb	81.00%	96.00%	N/A	78.31%	**98.59%**	97.51%	97.71%	98.00%	97.69%
Average	65.96%	**95.34%**	N/A	75.71%	90.21%	91.81%	90.05%	89.11%	92.30%

算块密度特征、标签路径特征、融合特征和 thresh 函数的设计分析可知, 其算法最坏时间复杂度分别为 $O(m)$, $O(m)$, $O(m)$ 和 $O(n)$. 算法 CEDP-NLTD 的最坏时间复杂度为 $O(n \times (f_{\text{densityF}}(m) + f_{\text{pathF}}(m) + f_{\text{combine}}(m) + f_{\text{thresh}}(n))) = O(mn)$. 通过对 CETD 的 TD, CTD 和 DSum 特征设计分析后可知, CEDP-TD, CEDP-CTD 和 CEDP-DSum 的最坏时间复杂度也为 $O(mn)$. 图 1 给出了各个算法的平均抽取时间性能对比, 发现这几个算法的平均抽取时间均在 90~100 ms 之间, 性能相差无几, 均可满足实时在线抽取任务的要求.

6.3 问题与讨论

本文提出的 CEDP 模型具有良好的可扩展性, 通过设计不同的密度函数可以得到不同的内容抽取算法. 然而, CEDP-TD, CEDP-CTD, CEDP-DSum, CEDP-NLTD 的性能有较大的差异, 说明 CEDP 模型受密度函数设计的影响较大, CEDP 模型取得好的抽取效果需要使用人工精心设计的密度函数. 另外, 从表 1 可以发现, 本文设计的 NLTD 密度函数适用于 CEDP 模型, 优于直接使用 CETD 提出的 3 种密度函数的效果. 基于 CEDP 模型设计更好的密度函数是后期工作需进一步研究的问题.

尽管信息融合技术已经成功运用于多种领域, 但目前在理论上尚未形成统一的框架, 在实际系统设计上缺乏有效的指导原则. 在学术界一直存在哪种算法最优的讨论, 但在选择信息融合算法的时候, 很难有完美的算法. 通常是根据可以获得的先验数据来分析和选择适当的鲁棒和精确的算法, 这也带来了融合算法对先验信息的依赖问题 [32, 33]. 本文尝试了块密度加权的标签路径特征, 显然, 这仅仅是其中一种融合方法, 将具有坚实数学基础的融合原理和方法应用到本文所提模型也是需要进一步研究

表 3　每个算法在各个数据集上的抽取召回率 (优胜者标记为粗体)

Table 3　Recall for each algorithm in each source (the best sources are marked in bold)

DataSet	CETR	CEPR-AT	CETD-QT	CETD-Jsoup	CEPF	CEDP-TD	CEDP-CTD	CEDP-DSum	CEDP-NLTD
CleanEval-En	87.20%	68.00%	**94.93%**	93.18%	84.28%	83.88%	61.98%	63.78%	83.38%
CleanEval-Zh	87.48%	72.67%	N/A	**90.35%**	85.40%	85.33%	68.82%	69.05%	85.07%
NY Post	91.40%	71.43%	95.75%	**95.77%**	88.07%	87.28%	87.30%	87.33%	87.39%
Freep	90.93%	90.46%	**99.22%**	98.98%	96.29%	95.76%	95.19%	96.38%	95.65%
Suntimes	97.95%	78.25%	**99.18%**	98.95%	93.44%	93.55%	94.04%	94.03%	93.74%
Techweb	97.68%	87.31%	**99.67%**	99.27%	94.77%	95.13%	91.79%	94.47%	95.10%
Tribune	95.24%	83.62%	N/A	**98.64%**	94.12%	93.51%	93.56%	93.50%	93.58%
Nytimes	93.40%	78.29%	96.00%	**96.64%**	88.78%	87.74%	85.91%	85.30%	87.85%
BBC	92.17%	70.88%	**98.49%**	98.41%	87.41%	87.15%	86.67%	87.03%	87.01%
Reuters	92.75%	77.01%	N/A	**99.61%**	93.41%	93.44%	76.91%	77.49%	93.44%
Yahoo!	94.22%	78.86%	97.08%	**97.70%**	85.91%	88.66%	87.62%	88.98%	88.91%
Sina	**98.94%**	85.54%	N/A	98.47%	97.55%	97.09%	96.93%	97.16%	97.04%
People	97.89%	79.12%	99.70%	**99.91%**	94.64%	94.04%	93.16%	94.65%	93.83%
163	88.76%	81.46%	N/A	**89.93%**	87.24%	87.45%	87.31%	86.81%	87.49%
Xinhua	98.68%	73.94%	99.15%	**99.25%**	93.93%	93.22%	93.12%	93.41%	93.11%
TunxunWb	**98.66%**	10.25%	83.81%	85.44%	72.34%	79.27%	76.63%	77.21%	77.95%
SinaWb	94.56%	11.10%	N/A	95.70%	**97.61%**	94.94%	79.98%	97.59%	91.72%
SohuWb	94.33%	89.53%	N/A	**99.30%**	88.92%	87.47%	87.42%	87.37%	87.42%
Average	94.01%	71.54%	N/A	**96.42%**	90.23%	90.27%	85.80%	87.31%	89.98%

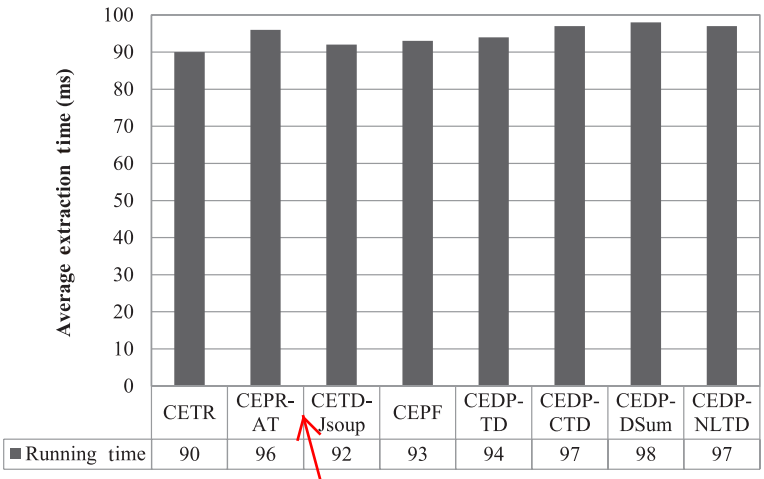

图 1　每个算法的平均抽取时间性能

Figure 1　Average extraction time of each algorithm

与正文表述内容相比，缺少CETD-QT。此图表是否正确？作者核。CETD-QT在8个数据集上无法正常运行下去（N/A）,因此，不能参加比较。图表正确，已核实。

的问题.

阈值函数 thresh(T) 中的超参数 α, β, δ 的设置[illegible]σ_T] 范围内搜索出最佳阈值. 不同超参数的影响表现在: α[illegible]阈值可

能会抽取较多不该抽取的内容, 在抽取性能上表现为较低的抽取精度、较高的召回率、较低的 F 值; β 越小, 越有可能求解出较大的阈值, 较大的阈值可能会漏抽较多该抽取的内容, 在抽取性能上表现为较高的抽取精度、较低的召回率、较低的 F 值. δ 越小, 搜索的粒度越细, 抽取性能对目标函数的设计越敏感, δ 过大则难以求解最佳阈值. 能否设计具有良好性能、含有尽量少超参数的阈值函数也值得进一步研究.

7 总结与展望

Web 内容抽取已有近二十年的研究工作, 但大数据开放环境下的实时在线抽取问题也是近年来才被研究者关注. 本文在密度特征和标签路径特征这两类具有良好抽取性能特征的基础上, 通过扩展标记有序树模型有机结合了这两类设计迥异的抽取特征, 分别发挥了密度特征在文本块级别区分内容和噪音的优势和标签路径特征在全局范围内区分内容和噪音的优势. 提出的结合文本块密度和标签路径特征的 Web 内容抽取模型具有良好的可扩展性, 设计的块密度加权标签路径特征的 Web 新闻抽取方法具有优良的抽取性能, 可满足大数据开放环境下的实时在线抽取任务的需求.

本文提出的模型和方法还存在改进空间. 能否设计更好的密度特征函数、标签路径特征函数、融合函数和阈值函数? 能否设计从不同角度区分内容和噪音的密度特征系, 并通过研究自适应的密度函数选择的方法或开展密度特征系的融合研究得到区分性能更好的密度特征? 能否通过协同抽取等策略深度融合密度特征和标签路径特征以提高抽取性能? 这些都是值得进一步研究的问题.

参考文献

1 CNNIC. Statistical report on Internet development in China. Technical report. China Internet Network Information Center, 2016. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201608/P020160803367337470363.pdf [CNNIC. 中国互联网络发展状况统计报告. 中国互联网络信息中心. 2016. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201608/P020160803367337470363.pdf]

2 Wu X, Zhu X, Wu G Q, et al. Data mining with big data. IEEE Trans Knowl Data Eng, 2014, 26: 97–107

3 Wu X, Chen H, Wu G, et al. Knowledge engineering with big data. IEEE Intell Syst, 2015, 30: 46–55

4 Li X L, Gong H G. A survey on big data systems. Sci Sin Inform, 2015, 45: 1–44 [李学龙, 龚海刚. 大数据系统综述. 中国科学: 信息科学, 2015, 45: 1–44]

5 Glynn C J, Herbst S, Lindeman M, et al. Public Opinion. Colorado: Westview Press, 2015

6 Zhu C, Zhu H, Ge Y, et al. Tracking the evolution of social emotions with topic models. Knowl Inform Syst, 2016, 47: 517–544

7 Etzioni O, Fader A, Christensen J, et al. Open information extraction: the second generation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, 2011. 1: 3–10

8 Zhao J, Liu K, Zhou G Y, et al. Open information extraction. J Chinese Inform Proces, 2011, 25: 98–111 [赵军, 刘康, 周光有, 等. 开放式文本信息抽取. 中文信息学报, 2011, 25: 98–111]

9 Parapar J, Barreiro A. An effective and efficient web news extraction technique for an operational NewsIR system. In: Proceedings of the Conferencia de la Asociación Espanola para la Inteligencia Artificial CAEPIA-TTIA, Salamanca, 2007. 2: 319–328

10 Weninger T, Palacios R, Crescenzi V, et al. Web content extraction: a metaAnalysis of its past and thoughts on its future. ACM SIGKDD Explor Newslett, 2016, 17: 17–23

11 Chang C H, Kayed M, Girgis M R, et al. A survey of web information extraction systems. IEEE Trans Knowl Data Eng, 2006, 18: 1411–1428

12 Ferrara E, de Meo P, Fiumara G, et al. Web data extraction, applications and techniques: a survey. Knowl-Based Syst, 2014, 70: 301–323

13 Jiménez P, Corchuelo R. Roller: a novel approach to Web information extraction. Knowl Inform Syst, 2016, 49: 197–241
14 Wu G Q, Hu J, Li L, et al. Online web news extraction via tag path feature fusion. J Softw, 2016, 27: 714–735 [吴共庆, 胡骏, 李莉, 等. 基于标签路径特征融合的在线 Web 新闻内容抽取. 软件学报, 2016, 27: 714–735]
15 Sun F, Song D, Liao L. Dom based content extraction via text density. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, 2011. 245–254
16 Wu X D, He J, Lu Y Q, et al. From big data to big knowledge: HACE+BigKE. Acta Autom Sin, 2016, 42: 965–982 [吴信东, 何进, 陆汝钤, 等. 从大数据到大知识: HACE+BigKE. 自动化学报, 2016, 42: 965–982]
17 Xue Y, Hu Y, Xin G, et al. Web page title extraction and its application. Inform Proces Manage, 2007, 43: 1332–1347
18 Zhao X, Jin P, Yue L. Discovering topic time from web news. Inform Proces Manage, 2015, 51: 869–890
19 Garcia-Molina H, Hammer J, McHugh J. Semistructured data: the TSIMMIS experience. In: Proceedings of the 1st East-European Conference on Advances in Databases and Information systems, St Petersburg, 1997. 1–8
20 Sahuguet A, Azavant F. Building intelligent web applications using lightweight wrappers. Data Knowl Eng, 2001, 36: 283–316
21 Liu L, Pu C, Han W. XWRAP: an XML-enabled wrapper construction system for web information sources. In: Proceedings of the 16th International Conference on Data Engineering, San Diego, 2000. 611–621
22 Wu G, Wu X. Extracting web news using tag path patterns. In: Proceedings of IEEE International Conference on Web Intelligence and Intelligent Agent Technology, Macau, 2012. 1: 588–595
23 Wu S, Liu J, Fan J. Automatic web content extraction by combination of learning and grouping. In: Proceedings of the 24th International Conference on World Wide Web, Florence, 2015. 1264–1274
24 Dalvi N, Bohannon P, Sha F. Robust web extraction: an approach based on a probabilistic tree-edit model. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, Providence, 2009. 335–348
25 Parameswaran A, Dalvi N, Garcia-Molina H, et al. Optimal schemes for robust web extraction. In: Proceedings of the VLDB Endowment, Seattle, 2011. 4: 980–991
26 Hogue A, Karger D. Thresher: automating the unwrapping of semantic content from the World Wide Web. In: Proceedings of the 14th International Conference on World Wide Web, Chiba, 2005. 86–95
27 Alarte J, Insa D, Silva J, et al. TeMex: the web template extractor. In: Proceedings of the 24th International Conference on World Wide Web, Florence, 2015. 155–158
28 Wu G, Li L, Hu X, et al. Web news extraction via path ratios. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, San Francisco, 2013. 2059–2068
29 Peters M E, Lecocq D. Content extraction using diverse feature sets. In: Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, 2013. 89–90
30 Otsu N. A threshold selection method from gray-level histograms. Automatica, 1975, 11: 23–27
31 Baroni M, Chantree F, Kilgarriff A, et al. Cleaneval: a competition for cleaning web pages. In: Proceedings of the International Conference on Language Resources and Evaluation, Marrakech, 2008. 638–643
32 Pan Q, Yu X, Cheng Y, et al. Essential methods and progress of information fusion theory. Acta Autom Sin, 2003, 29: 599–615 [潘泉, 于昕, 程咏梅, 等. 信息融合理论的基本方法与进展. 自动化学报, 2003, 29: 599–615]
33 Pan Q, Wang Z F, Liang Y, et al. Basic methods and progress of information fusion (II). Control Theory Appl, 2012, 29: 1233–1244 [潘泉, 王增福, 梁彦, 等. 信息融合理论的基本方法与进展 (II). 控制理论与应用, 2012, 29: 1233–1244]

Online Web news extraction via tag path feature weighted by text block density

Gongqing WU*, Pengcheng LIU, Jun HU & Xuegang HU*

School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009*, China*
* Corresponding author. E-mail: wugq@hfut.edu.cn, jsjxhuxg@hfut.edu.cn

Abstract Web news extraction is the basis of many "big data" and "big knowledge" applications, and it is also an open research problem. The tag path feature and the text block density feature are two excellent features to solve this problem at present. The tag path feature can well distinguish the content and noise of the whole webpage, but it is difficult to distinguish noise in the content block and the content in the noise block; the text block density feature can well recognize the content block with high density, but it is not robust enough. Aiming at the above mentioned problems, we propose a Web information extraction model, CEDP, which can effectively combine the tag path feature and the text block density feature, design a tag path feature weighted by the text block density to utilize the merits of the two features above, and design a Web news extraction method via the weighted tag path feature, CEDP-NLTD. CEDP-NLTD is a fast, universal, no-training and online Web news extraction algorithm, and it is suitable for extracting heterogeneous Web news in the Web big data environment across multi-resources, multi-styles, and multi-languages. Experiments on public data sets such as CleanEval show that the CEDP-NLTD method achieves better performance than the state-of-the-art CETR, CETD, CEPR and CEPF methods, and achieves better performance than CEDP-TD, CEDP-CTD and CEDP-DSum, which are generated from CEDP by using one of three block density features of CETD.

Keywords content extraction, Web news, text block density, tag path feature, online algorithm

Gongqing WU was born in 1975. He received a Ph.D. degree in 2013 from Hefei University of Technology. He is currently an associate professor at Hefei University of Technology. His research interests include data mining and Web intelligence.

Pengcheng LIU was born in 1991. He is currently a graduate student for a Master's degree at the School of Computer Science and Information Engineering, Hefei University of Technology. His research interests include Web data integration and data mining.

Jun HU was born in 1990. He received a master's degree in 2016 from Hefei University of Technology. He is currently a Ph.D. student at HeFei University of Technology. His research interests include web intelligence and multimodal deep learning.

Xuegang HU was born in 1961. He received a Ph.D. degree in 2000 from Hefei University of Technology. He is a professor at Hefei University of Technology. His research interests include data mining and machine learning.