文章编号:1001 - 9081(2009)02 - 0403 - 03

关于重复词提取的两种算法分析

蒋 华,殷 波
(桂林电子科技大学 计算机与控制学院,广西 桂林 541004)
(jianghua0773@126.com; raul_yinbo@163.com)

摘 要:针对重复网页的去重问题,对两种重复词句提取算法进行了系统分析比较。STC 算法在时间成本上具有优秀性能,重复序列的倒排索引方法在空间复杂度方面更胜一筹。结合 STC 算法对重复序列方法进行了改进,而面向主题转载的重复网页,先抽取重复串,然后将重复串作索引进行 STC 算法的重复抽取。实验结果表明,改进算法在保持了原有空间特性的基础上极大地提高了时间效率。

关键词:重复词句;重复序列;后缀树
中图分类号: TP391.1 **文献标志码**:A

New algorithm based on repeat sequence deletion

JIANG Hua, YIN Bo
(*Department of Computer and Control, Guilin University of Electronic Technology, Guilin Guangxi* 541004, *China*)

Abstract: Aiming at the current de-duplication algorithms, two repeated sequences (RS) extracting algorithms were compared and analyzed. Since STC has favorable performance in considering time cost and the inverted index method is superior in terms of spatial complexity, STC was used to improve RS algorithm. Experiment results show that this method can find similar Web pages efficiently. This algorithm can reach a high precision in mono-language deletion of duplicated Web pages, and this algorithm can also reach a maximum precision when it is applied to deletion of duplicated web pages.

Key words: repeated sequences; repeated segments; suffix tree

0 引言

重复词句字段,短语,搭配等的重复或同现,有时是指重复序列,重复语句,同现的字句,或重复排列等,更一般而言,它是在一篇文本或文集中出现超过两次的字序列。在文本中所有的字都考虑或者通过 stop list 忽略文中某些词,都是随着具体实现而定。一般来说,进行重复判断时可把某些词看作特定符号来标记,这样重复字句的符号标记就和原始文本中的样式一样。重复及相似内容的识别在信息处理领域,文本信息提取领域,都是重要的研究课题,它还广泛应用于防抄袭识别、新闻网页去重、自动分类、搜索引擎等系统中。重复字段的抽取和实现涉及和涵盖了计算语言学和文本挖掘的范围,它们对聚类,分类,主题发现和其他机器学习和人工智能技术等都有着重要意义。近年来更广泛的应用于字符串处理,DNA 序列比对,文本聚类,XML 结构索引等领域中。对重复词句提取算法的研究有着重要的现实意义。

1 后缀树

后缀树(Suffix Tree,ST),把文档看作是一个由若干短语组成的字符串,而不是看作一组词集[1]。一般算法中用来找出重复语句的传统 ST 结构都类似于 Zamir[2] 中阐述的结构。在最简单的版本中,后缀树算法创建了一个树型结构,每个后缀树的节点,表示一个词并且根节点为空。它是一个有根树,也是一个有向树。我们用下列语句来生成一个后缀树结构。

"can drive trucks safely. Men drive cars safely. Men can drive trucks."

首先确定树的最大深度(这样最大语句长度就被确定了) $m=3$。如图 1 生成了树形结构后,获得一个单一重复语句的最简单方法是用递归方法从根节点遍历到叶节点。如果使用非递归执行来获得语句,是一个相对复杂的过程,并且由于非递归方法取决于编译执行器的执行效率,因此在相同复杂度的情况下非递归不能保证更好的效率。在后缀树最外层描述一个句子的最后一个词的节点中常给出一个句子的出现频率[3]。

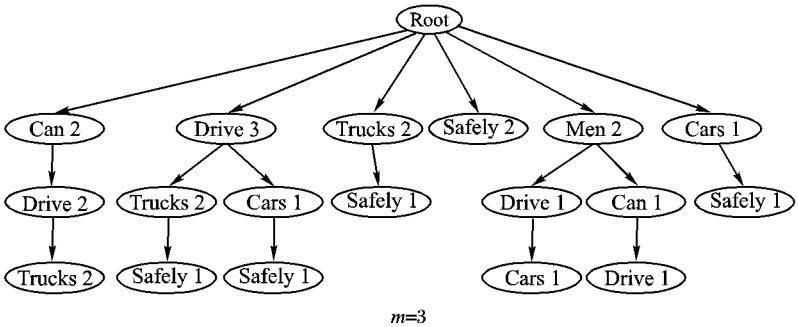

图 1 样本的后缀树结构

1.1 STC 算法分析

首先,一个输入文集被存储在分表中,以一种方式使特定的词被连续存储,并且句子或由标点分开的连续句子在空间上划界。接下来生成一个根节点。这个根节点的特殊性在于它不能指向任何一个字词。然后分表的第一个位置的词被读出,并被添加到有关键码值的哈希表中,哈希表[4] 是用来加快存储字句的存取。

收稿日期:2008 - 09 - 02;**修回日期**:2008 - 10 - 27。 **基金项目**:桂林电子科技大学博士科技基金资助项目(Z206116)

作者简介:蒋华(1963 -),男,河南信阳人,副教授,博士,主要研究方向:数据库、信息安全; 殷波(1984 -),女,山东聊城人,硕士研究生,主要研究方向:数据库、数据挖掘、信息安全。

每个后缀树节点表示输入文集的一个字词,并且它们平行生成一个对应哈希表的接口。这样如果一个后缀树中更深层的节点重复,就能减少总的存储空间。如果一个字词,在分表中被读出并写入到哈希表中,根节点应检测:当它是一个父节点时它是否包含一个节点表示当前的字词。如果是,则出现次数的计数应当累加;否则,生成一个表示当前字词的新节点并且把它添加到根节点的孩子节点的列表中。在这两种情况下,表示当前字词的节点都被添加到了父节点列表的末尾。接下来,下一个字词被读出并且这个过程如上述重复进行,直至分表读完。

图2描述了生成后缀树结构的四个步骤,输入句子是"can drive trucks safely",最大层次 $m = 3$。

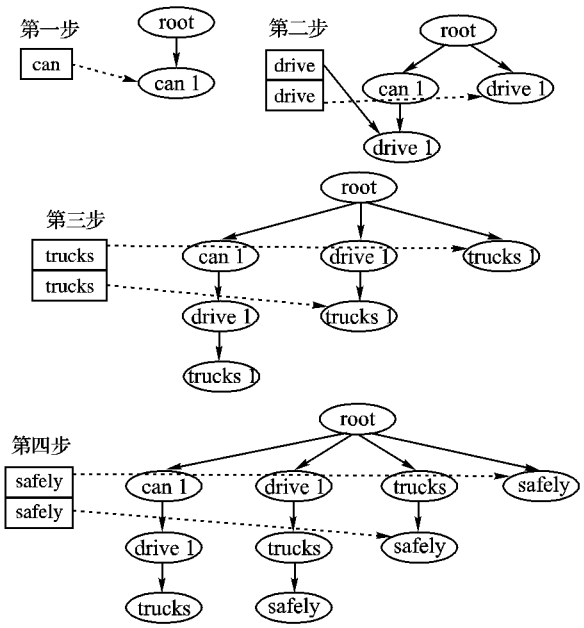

图2 后缀树结构解释实例 $m = 3$

1.2 复杂度

后缀树算法的时间复杂度是 $O(N + N*m + N)$,空间复杂度是 $O(N)$,有关键码值的哈希表是 $O(K)$。其中 N 是所有输入文集中字词的数目,m 是后缀树的最大层次,K 是在输入文集中关键词的个数。最坏的情况是包含所有后缀树节点的情况,其时间复杂度为 $O(N*m)$;对所有的 ST 语句而言,总的时间复杂度为 $O(N*m)$。一般而言,总的时间和空间复杂度是线性的。

2 重复序列的改进算法

2.1 重复序列

关于重复序列,文献[5,6]对它有特定的含义并且在定义上有着严格的限制。这种意义上的重复序列是指它由几个词组成并且有一个特定的含义。于是为了发现重复序列就需要增加额外的资源,如不同的过滤器等。对于这种严格定义的重复序列来说,它本身基于语义有着严格的语法特性,不具有语言独立性。

后来发展出了 LIKES 工具。它使用两个过滤器,一个切割过滤器,用来过滤一串不能组成其他队列的单独的词,多为关联词和动词;另一个语法过滤器,用来处理一串可以引导一个序列的代名词。这些过滤器有着语言依赖性。LIKES 能创建一个关于整个文档各个段、句子、词的树。可以找到一个序列在文中的特定位置。

2.2 算法分析

STC 算法合并高度重叠的基本类。优点是不必人为指定聚类类别的数目。利用每个类中各个文本包含的共同索引短语来描述这些类。STC 算法有较高的效率和线性的时间复杂度。但 STC 算法并非基于主题的查重,由于很多网页转载并非原文转载,而是稍作更改的重复转载,故 STC 算法无能为力。本文提出的改进算法,面向主题转载的重复网页,先抽取重复串。后将重复串作索引进行 STC 算法的重复抽取,关键重复串则是有着语义结合意义的短语串。这样做至少有两方面的好处:1)通过充分发掘文档所提供的信息可以改进聚类质量。2)有助于为产生的结果类提供简洁、易于理解的标签。针对 BOW 算法的不足,提出短语标引即使用短语作为标引项,短语表示词之间的临近关系及次序等,有更强的描述力。

一个重复串就是在一定数量文档中重复、频繁出现的一组词语,它可以描述属于这些文档的共同属性。我们从语料中提出重复串用作文档特征,显著降低了文档数据的维度,有效解决了文本聚类的"高维诅咒"同时可以为聚类结果提供简洁明了的类别标签。

2.3 算法描述

输入:文档集合 $D = \{d_1, d_2, \cdots, d_n\}$,其中 n 为全体文档数目;Θ: 属性值 MI 的阈值;γ: 属性值 IND 的阈值。

输出:文档特征向量集合:

$V = \{v_1, v_2, \cdots, v_n\}$, $V_i = \{(t_1, f_1), \cdots, (t_2, f_2)\}$, $1 \leqslant i \leqslant n$ 其中 t 表示文档 d 的一个特征,f 是 t 在 d 中出现频度。

步骤:

1)语料预处理。扫描 D 中每个文档,去除停用词,转换成内部表示。

2)文档解析,重复串发现及相关属性计算。

3)文本特征提取。对于2)中得到的重复串,根据完整性、稳定性和独立性,滤去属性值低于给定阈值的,余下的选作特征。

4)根据选中的特征,构建文档特征向量集合 V。

5)将重复串作为节点生成后缀树,进行 STC 算法重复抽取。

图3 重要的重复序列子句的发现过程

3 实验结果及分析

3.1 实验数据

本文在三个长度为大约 1 MB 到 30 MB 的不同文本上运行这两个算法,这些文本中字词的总数大致为几百万个。本文统计了字句最大长度为 3,4 和 5 的句子,并且统计了每个算法的时间和空间复杂度。在每个文本运行这两种算法得到的抽取重复词大致相同。图4、5 中的曲线证实了本文对于对算法的复杂度的理论假定。STC 算法在时间和空间上都是线性的,而 RS 算法在时间和空间复杂度上则是线性的。STC 算

法的空间消耗大致是 RS 算法空间消耗的两倍。两个算法的运行环境都是 C#,使用. NET Framework2.0 编译实现 AMD Opteron 1.6 GHz 2 GB RAM。

3.2 结果分析

通过分词和本文提出的特征提取方法对 3 个语料集进行文档标引到的不同特征数目。基于重复串的特征提取方法能有效降低特征空间的维度。在语料 Forum-14 上,相对于分词,特征词条数目减少了 70%。

对具有 30 MB 大小的英文文档进行实验。两种不同方法的时间和空间复杂度分别如图 4、5 所示。

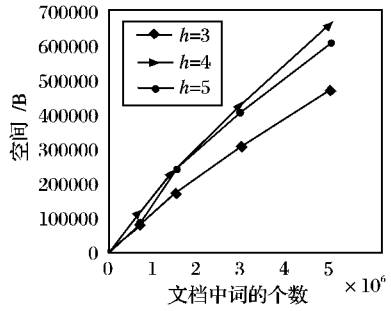

图 4　RS 空间复杂度

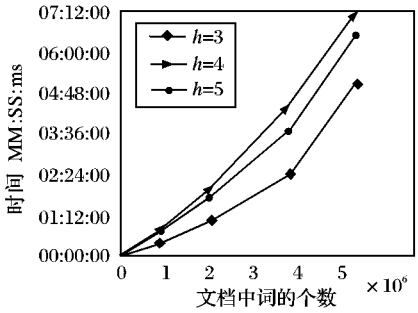

图 5　RS 时间复杂度

曲线证实了我们对算法的复杂度的理论假定。从图 4、5 明显看出,本文改进算法在保持原来空间特性的基础上使时间效率得到了极大的提高。

4 结语

利用 STC 对重复序列方法时间效率的提高,本文提出了一种改进的算法,核心思想是对字符重复串进行抽取,使用重复串作为短语标引生成后缀树,并映射生成倒排索引进行 STC 算法去重。实验证实了改进算法有着良好的准确率和召回率,并有着优良的时间和空间特性。重复词提取在各个领域的广泛应用,对其的研究必将有着更广的现实意义。

参考文献:

[1] GROLMUS P, HYNEK J, JEZEK K. User profile identification based on text mining[C]// Proceedings of 6th International Conference on Information Systems Implementation and Modelling. Chicago: Amsterdam University Press, 2003: 109 - 118.

[2] ZAMIR O, ETZIONI O. Web document clustering[C]// Proceedings of the 21 st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New YORK: ACM Press. 1998: 24 - 28.

[3] DEBAR H, WESPI A, DACIER M. Fixed vs. variable-length patterns for detecting suspicious process[J]. Journal of Computer Security, 2000, 8(2): 159 - 181.

[4] HAN J, KAMBER M. Data mining[M]. San Fransisco: Morgan Kaufmann Publishers, 2000.

[5] JUSTESON J, KATZ M. Technical terminology: some linguistic properties and an algorithm for identification in text[J], Natural Language Engineering, 1995, 1(1): 9 - 27.

[6] LEBART L, SALEM A. Statistique textuelle[EB/OL]. [2008 - 06 - 25]. http://ses.telecom-paristech.fr/lebart/ST.html.

(上接第 402 页)

图形的构造和电子几何文本的生成,几何知识库的测试数据主要选自文献[3 - 5,7]。目前,数据库中包含 44 条欧式初等几何概念,300 多条欧式初等几何定理、公理和问题。这些数据已按照前文所述的标准化和结构化格式存储,包含几何知识对象的自然语言描述、形式化描述、代数形式、几何形式等。

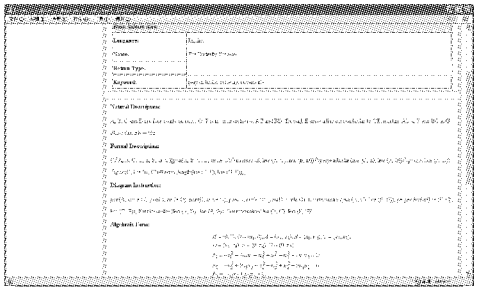

图 9　数据查询界面

5 结语

本文主要研究了几何知识数据的封装以及几何知识对象之间的关系,进而探讨了几何知识数据的标准化和结构化问题,讨论了几何知识库的设计与实施,最后介绍了我们已经实现的几何知识库系统的初级版本。今后的研究工作主要包括两个方面:一方面是逐步改进几何知识库的结构设计,使其对应用更加合理;另一方面是设计几何知识数据的处理和应用算法,实现本文所述的几何知识库系统的自动化功能,例如,作图语句的自动生成、形式化描述到自然语言描述和代数表示的自动翻译与转换等。

参考文献:

[1] Apache Jakarta Project. Apache software foundation [EB/OL]. [2008 - 06 - 15]. http://jakarta.apache.org/tomcat.

[2] Apache Jakarta Project. Apache software foundation [EB/OL]. [2008 - 06 - 15]. http://www.apache.org/dist/jakarta/struts/.

[3] CHOU S C. Mechanical geometry theorem proving [M]. Boston: Springer, 1988.

[4] CHOU S C, GAO X S, ZHANG J Z. Machine proofs in geometry: Automated production of readable proofs for geometry theorems [M]. Singapore: World Scientific, 1994.

[5] COXETER H S M, GREITZER S L. Geometry revisited [M]. Washington, DC: Yale University, 1967.

[6] GRABE H G. The symbolicdata GEO records: A public repository of geometry theorem proof schemes[EB/OL]. [2008 - 06 - 15]. http://www.symbolicdata.org/Papers/linz-02.pdf.

[7] Euclid'sElements[EB/OL]. [2008 - 06 - 15]. http://aleph0.clarku.edu/~djoyce/java/elements/toc.html.

[8] Microsoft Corporation. Microsoft SQL server[EB/OL]. [2008 - 06 - 15]. http://www.microspft.com/sql.

[9] QUARESMA P. GeoThms-Geometry framework[EB/OL]. [2008 - 06 - 15]. http://www.mat.uc.pt/~pedro/cientificos/Publicacoes/techRepCISUC0602.pdf.

[10] WANG DONGMING. GEOTHER 1.1: Handling and proving geometric theorems automatically[C]// Automated Deduction in Geometry. Berlin Heidelberg: Springer-Verlag, 2004: 194 - 215.

[11] 周定康,许婕,李云洪,等. 关系数据库理论及应用[M]. 武汉:华中科技大学出版社,2005.