第 33 卷 第 2 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.2 Computer Engineering January 2007
·软件技术与数据库· 文章编号：1000—3428(2007)02—0065—03 文献标识码：A 中图分类号：TP312

重复串特征提取算法及其在文本聚类中的应用

胡吉祥[1,2]，许洪波[1]，刘 悦[1]，程学旗[1]
(1. 中国科学院计算技术研究所，北京 100080；2. 中国科学院研究生院，北京 100039)

摘 要：针对 Web 文档的高维问题及网络新语言给现有分词系统带来的挑战，该文提出一种基于重复串的特征提取方法，可以从文本中提取有意义的特征，且对于中文无需分词。实验表明，该方法可以降低特征空间维度，同时能有效改善传统以词为特征的聚类算法的性能。
关键词：文本聚类；特征提取；重复串

Algorithm of Repeats-based Term Extraction and Its Application in Text Clustering

HU Jixiang[1,2], XU Hongbo[1], LIU Yue[1], CHENG Xueqi[1]
(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080;
2. Graduate School, Chinese Academy of Sciences, Beijing 100039)

【Abstract】This paper proposes a novel term extraction method based on repeats, which can extract meaningful terms from text. For Chinese, it need not word segmentation. Experimental results show that the proposed approach can remarkably reduce the dimensionality and effectively improve the performance of traditional clustering algorithms.
【Key words】Text clustering; Term extraction; Repeats

文本聚类分析是组织文档的一种有效手段，它被广泛应用于检索、过滤和分类 Web 上的文档。与文本分类不同，在文本聚类分析中没有文档类别信息可利用。尽管许多传统聚类技术如 K-Means 等可以应用于文档聚类，但它们都难以满足 Web 文档聚类的特殊需求：数据的高维度，大数据量，易于浏览的聚类结果，以及清晰简洁的类别标签。

文本聚类的首要工作是文档标引，即将文本表示成机器内部可计算的形式。传统 IR 中采用的是 BOW(Bag-Of-Words)方法，将文档表示为词及其在该文档中出现频度的一个向量。它仅简单地考虑一个词是否在文档中出现及其出现频度，而忽略了词出现的词法、语法及实际上下文语景。文本聚类的另外一个主要的困难是“高维诅咒”问题。即使是一个中等大小的文档，由单个特征所组成的原始特征空间也可能高达成百上千维。这对于许多机器学习算法而言，其代价是高昂的。高维问题极大降低了分类和聚类算法的性能。随着文本数量的急剧增长和数据维度的不断增大，这个问题变得更加严重。

针对以上问题，本文提出一种新颖的基于重复串(Repeats)的文本特征提取方法。其基本思想是文档集合中谈论同一个话题的文档往往会包含许多共同的或相似的词语。因此同一个主题的文本中会出现大量相似的重复短语，这些重复短语正是构成文本之间相似性的重要因素。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，同时可以为聚类结果提供简洁明了的类别标签。此外，对于中文文档，该算法无需分词即可提出有意义的文本特征。实验表明，该算法应用于文本聚类中取得了比传统基于词的聚类方法更好的结果。

1 相关工作

由于BOW方法的不足，许多研究者提出了短语标引，即除了单个的词外，使用短语作为标引项。与单个的词相比，短语包含更多的信息，如单词之间的邻近关系以及次序等，而且有更强的描述能力。根据产生方式的不同，通常短语分为两种：词法性和统计性的。其中统计性短语被证明在IR、分类、聚类[1]中都能提高性能。一种典型的统计性短语是基于互信息度量的n元语法。然而，n元语法标引的不足在于即使当n取值很少时，如 1,2 或 3，其时空代价也很高。文 献[2]的实验结果表明，当n元语法的长度取值为 2,3 或 4 时，分类性能相对于使用单个词作为标引项时有显著提高；而当n取值到 5 或更大时，使用n元语法反而会降低分类性能。

本文提出一种基于重复串的特征提取方法，使用在文本中频繁出现的短语作为特征。这些短语既具有相对完整的语义，同时也有较好的统计特性。本文提出的特征提取的方法也是基于统计的。由于对 n 的取值并不固定为某个值，它可以被视作 n 元语法的一个泛化模型。与 n 元语法相比，它采用了不同的提取方法，其复杂度相对较低，同时提取的特征可以为任意长度，而不影响分类或聚类的性能。

2 基于重复串的特征提取算法

大多数聚类算法将一篇文档仅仅视为一组词的集合，完全忽略了词语之间的顺序及共现关系，而这些可能为文档聚类提供重要信息。因此，我们从全体文档集合中抽取关键重

基金项目：国家“973”计划基金资助项目(2004CB318109)
作者简介：胡吉祥(1983－)，男，硕士生，主研方向：信息检索和文本挖掘；许洪波，博士、副研究员；刘 悦，博士、助理研究员；程学旗，博士、研究员
收稿日期：2006-03-21 **E-mail：**hujixiang@software.ict.ac.cn

复串作为文档特征。一个重复串被定义为一篇文档中一个子串，它是由一个或者更多的词组成的有序序列。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。

2.1 基本定义

给定一个长为 N 的文档串 S，我们通过以下 3 个属性来判定 S 的一个子串 R 是否是一个关键重复串：完整性，稳定性和独立性。

定义 1 假定一个子串R在串S中出现在k个不同的位置$p_1,p_2,\ldots,p_k$，R被称为一个最大化的重复串当且仅当它满足以下条件：至少存在一对i, j (1≤i＜j≤k，使得S的第(p_i-1)个字符和第 (p_j-1)个字符不相同，此时称R为左最大化的；或者至少存在一对i, j (1≤i＜j≤k，使得S的第(p_i +|R|)个字符和第(p_j +|R|)个字符不相同，此时称R为右最大化的。我们称同时左最大化和右最大化的重复串是完整的。

从以上定义中，不难推出以下引理：

引理 1 给定一个串 S，令~S 表示其逆串，即~S=S[n]S[n-1]…S[1]。则称 S 中一个子串 R 在 S 是左最大化的当且仅当其~R 在~S 中是右最大化的。

为了快速抽取串中的大多数重复串，最大化重复串的引入是必要也是足够的。它能够很简洁地捕获串的所有有意义的重复结构，同时还可避免产生大量无谓的输出：非最大化的重复串无需报告，因为它们必定包含于某些最大化的重复串中。但是，并非所有最大化的重复串都是有用的，很多只是部分短语，本身是语义不完整，无意义的，因此还需进一步过滤以筛选出其中有意义的、我们感兴趣的部分。

定义 2 给定输入串S=c1c2…cp，定义其稳定性为模式S的互信息，即 $MI(S) = f(S)/(f(S_L) + f(S_R)-f(S))$，其中 $S_L = c_1 \ldots c_{p-1}$, $S_R = c_2 \ldots c_p$，f(S), $f(S_L)$, $f(S_R)$分别是S, S_L, S_R的出现频度。显然f(S)要小于$f(S_L)$和$f(S_R)$。

互信息量 MI 比较了一个模式串及其部分子串的频度，它可以衡量模式串各部分之间的相关度。当一个模式串 S 及其子串共现频度高时，MI(S)较高且接近于 1，此时模式串 S 与单独其左、右部分子串相比更有可能形成一个短语。相反，如果 MI(S)较低且接近于 0，表明 S 不太可能形成一个短语。

定义 3 文献[3]指出，一个模式是独立的当其上下文的熵值很高，即其上下文均有足够的随机性。我们用 IND 来度量一个模式串的上下文独立性。下面的公式用于计算一个短语串 S 的上下文的独立性：

$$IND_l = -\sum_{\partial=lc(S)} \frac{f(\partial S)}{f(S)} \log \frac{f(\partial S)}{f(S)}$$

$$IND_r = -\sum_{\beta=rc(S)} \frac{f(S\beta)}{f(S)} \log \frac{f(S\beta)}{f(S)}$$

其中α、β 分别是串 S 的左右邻接串，α S、β S 表示两个串的连接。同时定义 0 · log0 = 0。最后的 IND 值为二者的平均值。IND 值越大，表明该模式的上下文独立性越高，即越可能形成一个短语。

通过完整性、稳定性和独立性 3 个属性，可以从文本中提取关键的重复串，同时避免了重复的子模式，以及无意义的部分短语。

2.2 算法描述

算法 1 基于重复串的特征提取算法

输入：文档集合 $D = \{d_1, d_2, \ldots, d_n\}$，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：

$V = \{v_1, v_2, \ldots v_n\}, v_i = \{(t_1, f_1), \ldots, (t_k, f_k)\}, 1 \le i \le n$

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 2.1 节中的 3 个属性，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。

算法先扫描语料中的每个文档，去除停用词以及数字标点等非单词符号。一个文档视作一个字符串，全体文档被连接成一个伪文档。每个单词被转换成一个 2 字节大小的整数，这样每个英语单词或中文汉字都可以被作为一个单元处理。同时，记录串中每个下标对应字符所属的文档编号，文档之间用特定的边界符号分隔，该边界符号不会出现在任何原始文档中。显然，跨越文档边界的子串是没有意义，我们限定算法发现的重复串在一个文档内。更为严格的是，由于句子边界往往意味着话题的转换，重复串也可被限定在一个句子内。这样也降低了重复串发现算法的代价。预处理的输出结果是一个包含语料集中全体文档内容的一个字符串，以及相应的文档编号记录。

由 2.1 节的定义可知，每个重复串的属性计算依赖于该模式及其子模式的频度统计信息。一个长为 N 的文本串有 N(N+1)/2 个子串，为了高效地获取所有子串的频度，引入文献[4]中提出的子串类概念。一个子串类是满足以下 3 个性质的子串集合：

(1)同一类中子串有相同的集合频度(cf)和相同的文档频度(df)；

(2)所有子串类构成全体子串的一个划分，即每个子串属于一个类且仅属于一个类；

(3)有 N 个子串其 cf=1 同时最多有 N-1 个子串其 cf>1。

通过子串类，所有子串被聚成相对少的类，只需计算至少 2N-1 个子串的的频度就可以得到所有子串的统计信息。所用到的数据结构是根据输入文本串所建立的后缀数组及相应的最大公共前缀 LCP 数组，其时间和空间复杂度分别为 O(NlogN) 和 O(N)[4]。此部分算法输出所有的右最大化重复串及其频度统计。

一个完整的重复串应当同时是左最大化和右最大化的。根据引理 1，为发现文档 T 所有左最大化重复串，只需将上述算法应用于其逆转串~T。如果 S 是~T 中一个右最大化子串，则~S 必定是 T 中的一个左最大化子串。通过对所有的左最大化子串集合和右最大化子串集合求交集，就可以得到文档 T 中所有完整的重复串及统计信息。设左最大化和右大化的子串数目分别为 L 和 R，则此部分算法的时间和空间复杂度均为 O(L+R)。由于 L<<N, R<<N，因此整个特征提取算法的时间和空间复杂度仍然是 O(NlogN) 和 O(N)。

在得到文档 T 中的完整重复串及其频度后，很容易计算出每个重复串的稳定性和独立性。拥有较高的 MI 和 IND 值的重复串被认为是更有可能形成一个有意义的短语。分别为 MI 和 IND 值设定两个阈值Θ和γ，属性值低于阈值的重复串被滤去。最后剩下的则正是我们所需要的完整、稳定和独立

的重复串，即关键重复串。

3 实验及结果分析

3.1 数据集

实验中使用了 3 个中文语料集。一个是 TDT2 语料中的中文部分，包括 3 208 篇文档，分为 20 个类，总词数为 25 479。TDT2 语料是从新华网，早报网和美国之音中文频道节目中收集的，时间跨度为 1998 年 1 月到 6 月。

另一个中文语料是由 Songbo Tan 收集整理的。该语料分为两个层次，第一层包含 12 个大类，每二层包含 60 个小类。总共有 14 150 篇文档，总词类约为 70 000。该语料可以被用作 3 个分类语料：一个层次数据集 (TanCorpHier) 和两个平坦数据集(TanCorp-12 and TanCorp-60)，在实验中我们采用的是 TanCorp-12。

第 3 个语料是我们从各大网站论坛上采集的，共 1 514 篇文档，平均大小为 80B。因为数据是来源于实际生活中，其包含的主题具有相当的不规范性，我们按照网站上的原始分类及自己的主观判断将其粗略划分为 14 个类别(主题)，每一个主题包含的文档数量不均衡，我们将该语料记为 Forum-14。

3.2 评价方法

对于聚类质量的评价，我们采用了一种常用的基于人工标注的评价指标F-measure[5]，它将每个聚类结果簇视为一个查询的结果，每个人工标注类视为一个查询的相关文档集合。对于每个人工类K_i和结果簇C_j，计算准确率、召回率和F-measure如下：

$$\Pr ecision\ (K_i, C_j) = \frac{|K_i \cap C_j|}{|C_j|}$$

$$\text{Re}\ call\ (K_i, C_j) = \frac{|K_i \cap C_j|}{|K_i|}$$

$$F(K_i, C_j) = \frac{2 * \Pr ecision\ (K_i, C_j) * \text{Re}\ call\ (K_i, C_j)}{\Pr ecision\ (K_i, C_j) + \text{Re}\ call\ (K_i, C_j)}$$

对每一个人工类K_i找出一个最能描述它的结果簇，即使$F(K_i, C_j)$最大的C_j。对一个聚类结果C，通过取它对所有人工类的最大F-measure的加权平均来衡量其聚类质量，该指标称为聚类结果C的总体F-measure，记为F(C):

$$F(C) = \sum_{Ki \in K} \frac{|Ki|}{|D|} \max_{Cj \in C} \{F(K_i, C_j)\}$$

其中K表示所有人工类，C表示所有结果簇；$|K_i|$表示类K_i中文档数目，|D|表示数据集中全体文档数目。F(C)取值范围是[0,1]，F(C)值越高表明聚类效果越好。

3.3 实验方案

为了对本文提出的方法进行有效的评估，这里设计了两个实验。

(1)使用在国际上广泛获得好评的中文分词系统ICTCLAS（中科院计算所汉语词法分析系统）对各个文本语料进行分词，再用几种典型的算法进行聚类实验。选取的聚类算法有：K-means，最近邻聚类(NNC)和Bisecting K-Means[5]共 3 种。

(2)采用前面提出的特征抽取算法进行采用相同的算法进行聚类实验。为排除无用噪声，舍去词频小于 2 的特征串及长度小于 2 的单字词。由于 K-means 算法的性能很容易受初始点的影响，对于每个数据集随机选择了 10 组初始点，取 10 次的平均值作为最后的聚类结果。

3.4 结果及分析

表 1 列出了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得到的不同特征数目。从表中可以看出，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 60%。

表 1 各语料上分词及重复串方法得到的特征数目对比

语料名称	文档数目	类别数目	特征数目	
			分词	重复串
TDT2	3 208	20	29 932	25 479
TanCorp	14 150	12	118 053	72 641
Forum-14	1 514	14	2 328	829

表 2 列出了采用 3 种聚类方法分别以词和重复串作为特征在 3 个语料集上聚类结果的 F-measure。可以看出，在采用基于重复串作为特征时，每种聚类算法的质量都有明显的提高。这充分说明了我们的方法是有效可行的，同时也验证了文献[1]中的结论。

表 2 基于分词和重复串特征的聚类方法 F-Measure 对比

语料名称	特征类型	聚类方法		
		K-Means	NNC	Bisecting
TDT2	分词	0.401	0.78	0.401
	重复串	0.426	0.886	0.403
TanCorp	分词	0.678	0.634	0.678
	重复串	0.689	0.661	0.741
Forum-14	分词	0.284	0.281	0.254
	重复串	0.483	0.427	0.447

4 结论与展望

聚类分析是文本挖掘中的一种重要手段，而文档标引和特征提取是其基础和关键工作。本文提出一种新的基于关键重复串的特征提取方法，可以从文本中提取有意义的重复串作为特征。该方法能降低特征空间维度，同时可有效改善传统以词为特征的聚类算法的性能。对于重复串特征的进一步精简和相关阈值的自动选择是我们下一步的研究工作。

参考文献

1 Zamir O E. Clustering Web Documents: A Phrase-based Method for Grouping Search Engine Results[D]. University of Washington, 1999.

2 Furnkranz J. A Study Using N-gram Features for Text Categorization[R]. Technical Report: TR-98-30, http://www.ai.univie.ac.at/cgi-bin/tr-online?number+98-30, 1998.

3 Chien L F. PAT-tree-based Adaptive Key Phrase Extraction for Intelligent Chinese Information Retrieval[J]. Information Process and Management, 1999, 35(4): 501-521.

4 Yamamoto M, Church K W. Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus[J]. Computational Linguistics, 2001, 27(1): 1-30.

5 Steinbach M, Karypis G, Kumar V. A Comparison of Document Clustering Techniques[C]. Proc. of KDD Workshop on Text Mining'00, 2000.

(上接第 64 页)

3 Christian R E. Managing Content with Directory Servers[D]. Karlsruhe, Germany: Karlsruhe University, 2000.

4 Wahl M, Howes T, Kille S. RFC2251-Lightweight Directory Access Protocol(V3) [EB/OL]. 1997. http://www.faqs.org/rfcs/rfc2251.html.

5 Hors A L, Hégaret P L, Wood L, et al. Document Object Model (DOM) Level 3 Core Specification Version 1.0[EB/OL]. 2004-04-07. http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407.

6 Clark J, DeRose S. XML Path Language (XPath) Version 1.0[EB/OL]. 1999-11. http://www.w3.org/TR/xpart.

7 熊曾刚，陈建新，张学敏. 关于 XML 数据库的研究与进展[J]. 情报杂志, 2005, 24(3): 43-47.