基于词或词组长度和频数的短中文文本关键词提取算法

陈伟鹤 刘 云
(江苏大学计算机科学与通信工程学院 镇江 212013)

摘 要 中文文本的关键词提取是自然语言处理研究中的难点。国内外大部分关键词提取的研究都是基于英文文本的,但其并不适用于中文文本的关键词提取。已有的针对中文文本的关键词提取算法大多适用于长文本,如何从一段短中文文本中准确地提取出具有实际意义且与此段中文文本的主题密切相关的词或词组是研究的重点。提出了面向中文文本的基于词或词组长度和频数的关键词提取算法,此算法首先提取文本中出现频数较高的词或词组,再根据这些词或词组的长度以及在文本中出现的频数计算权重,从而筛选出关键词或词组。该算法可以准确地从中文文本中提取出相对重要的词或词组,从而快速、准确地提取此段中文文本的主题。实验结果表明,基于词或词组长度和频数的中文文本关键词提取算法与已有的其他算法相比,可用于处理中文文本,且具有更高的准确性。

关键词 关键词提取,中文文本处理,音译词,网络新词
中图法分类号 TP391.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.009

Keyword Extraction Algorithm Based on Length and Frequency of Words or Phrases for Short Chinese Texts

CHEN Wei-he LIU Yun
(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract Keyword extraction for Chinese text is an important and difficult part of the text processing research, especially in the field of natural language processing research. Most existing studies focus on English text or long Chinese text, but due to their nature limitations, those keyword extraction algorithms can not apply to Chinese text. Those keyword extraction algorithms for English text are unsuitable for extracting keywords from Chinese texts. How to extract words or phrases accurately from Chinese text which are meaningful and closely related to the topics of this paragraph is the point of this paper. This paper presented a novel keyword extraction algorithm based on length and frequency of words or phrases for Chinese texts. This algorithm firstly extracts words or phrases with high frequency in the paragraph, then calculates the weight of the words or phrases according to the frequency and length of these words or phrases. Lastly, according to their weights, keywords are filtered out. This algorithm can extract the relative important words or phrases from the Chinese text accurately, which can help us find out the theme of this section efficiently and accurately. Experimental results show that compared with other keyword extraction algorithms, the proposed keyword extraction algorithm can process Chinese text with higher accuracy.

Keywords Keyword extraction, Chinese text processing, Transliterated words, Internet new words

1 引言

面对现今世界海量的文本信息,人们迫切需要一些自动化的方法来帮助自己从中快速发现真正需要的信息。因此自然语言处理[1],尤其是针对中文文本的处理,是研究的热点和难点[2,3]。

自然语言处理研究的内容十分广泛,在这些研究方向中,自动文摘、信息检索、文档分类、信息过滤、信息抽取以及舆情分析等都涉及一个重要的核心问题——关键词的提取[4]。因此,关键词的提取技术已经成为自然语言处理领域的研究热点。

中国文化源远流长,中文文本资料信息十分丰富,随着网络的普及和全球化的发展,网络文本中不断有新词涌现,各国文化相互传播,交流日益加深,众多的国外文献资料被翻译为中文,这些翻译文本中包含了很多音译词,以中文形式记录的各类信息知识充满了我们的生活。面对着这浩如烟海的信息资料,如何快速、准确地了解这些文本信息所要表述的主要内容,成为了亟待解决的问题。自动中文文本关键词提取技术应运而生,它能够从文本中快速提取出重要的词或词组,从而帮助我们快速地了解一段文本的主题。如何快速准确地提取出中文文本中的关键词是本文研究的重点。

到稿日期:2015-11-04 返修日期:2016-03-13 本文受国家自然科学基金项目(61300228),江苏省教育厅自然科学基金(09KJB520003)资助。
陈伟鹤(1974—),男,博士,副教授,CCF会员,主要研究方向为数据库安全、模型检测、数据挖掘,E-mail:chenweiheujs@aliyun.com;刘 云(1991—),硕士生,主要研究方向为数据挖掘。

1.1 中英文关键词提取方法的不同

英文文本下的关键词提取算法不适用于中文文本的关键词提取。众所周知，英文文本是以单词为单位，单词和单词之间是靠空格隔开，而中文文本是以汉字为单位，汉字是构成词或词组的要素，一定个数的汉字组成词或者词组[10]。例如，英文句子“We are Chinese”，用中文表达则为“我们是中国人”。计算机可以很简单地通过空格知道“Chinese”是一个单词，但是不会知道“中”、“国”、“人”3个字合起来才是一个具有实际意义的词组。因此，如何对中文文本进行准确分词或如何使得提取出来的关键词或词组没有长度上的限制，从而得到相对准确且具有实际意义的中文单词或词组，是中文文本下关键词提取的难点，也是本文研究的重点。

1.2 中文文本关键词提取的难点

由于网络的普及，信息的传播几乎都以文本的形式通过网络实现，例如各类网页、微博、豆瓣等，这些文本信息长度短，字符数一般在200～300之间，用于相互交流和分享知识，更新速度快，并且包含很多的新词[11]。

随着全球化的发展，各国之间交流日益频繁，语言之间的交汇日益增加，大量的外国文献资料被翻译成中文文本，外国文献资料中的地点名称、人物姓名和一些具有特定文化背景的单词等会用读音相近的汉字来表述，即为音译词。音译词(Transliterated Words)是以读音相近的字翻译外族语言而形成的单纯词，还有的是从外族包括国外其他民族和国内少数民族语言借来的词。

H. L. Mencken曾经说过：“任何一种活的语言像人一样会不断有少量血液流失，它最需要的是必须经常从别的语言获取新鲜血液。任何语言将自己大门关闭之日就是这种语言走向消亡之时”。音译词就是不同语言群体间接触的直接反映，是语言异质性构成的重要特征[12]。

所谓音译，即按音翻译，就是把一种语言的词语用另一种语言中跟它发音相同或近似的语码表示出来的翻译方法，其英文解释是“Transliteration refers to a transcription from one alphabet to another”[13]。

例如，“巴黎”是根据法语“Paris”的读音音译的，“Singapore”音译为“新加坡”，“さくら”是“樱花”，因为我们不常用日文输入法，所以在中文文本中也会使用它的音译词。

汉语中同一个发音会对应多个汉字，因此不同的人会有不同的写法，这也就要求中文文本下的关键词提取不被词典限制，从而将这些音译词提取出来。随着英语的普及，已经不只是在中文文档中将这些词音译为中文，这些词极有可能是这篇文档的主题词，因此跨语言提取关键词变得尤为重要。

正常情况下，在一篇文档中，若将一个外语单词音译为中文单词，一定会音译为同一个单词词组，不会出现两种写法，即在一篇文章中若将“Euclidean Distance”音译，只会出现“欧几里得距离”或“欧几里德距离”中的一种写法。此文档若是围绕“欧几里德距离”这个主题来写，则“欧几里德距离”这个词会在文档中多次出现，因此可以根据词在文档中出现的频数来判断词的重要性。同时，也要考虑是否有两个或两个以上的词在同一句话中多次同时出现，即单词共现。往往共现的单词更能代表一个段落和一篇文档的主题，比如：“数据挖掘”和“购物篮”这两个词组同时出现，可以推测这篇短文本讲述的主要内容是：数据挖掘中购物篮算法。

综上所述，从文本中提取出的词或词组的长度不定，就中文文本而言，词或词组的长度大于或等于2才具备实际意义，例如：由“聚类”、“文本挖掘”等词或词组可知，此段文本可能与数据挖掘中文本挖掘或者聚类算法方面的知识相关。因此提取出的词或词组长度越长，在文本中出现频数越高，就越具有代表意义；反之，如果提取出来的词的长度为1，即只提取一个汉字字符，例如：“类”、“掘”等，则没有太大的价值，这两个字可以引申出很多的意义，比如，此文本讲述的可能是与挖“掘”机相关的信息，也可能是程序设计中的“类”的相关信息，还可能是如何学习各“类”知识，等等，从而根本无法通过这些字推理出文本可能讲述的内容。因此，要想快速获取文本信息的主旨，准确、快速地提取中文文本中的关键词或词组十分重要。

2 相关工作

近年来，已经存在的对公共子串提取算法的研究中，比较有代表性的有最长公共子串算法[5,6]、N-Gram算法[7]、Hirschberg算法[8]和Nakatsu算法[9]。

最长公共子串(Longest Common Substring)和最长公共子序列(Longest Common Subsequence)的区别是：子串(Substring)是串的一个连续的部分，子序列(Subsequence)则是不改变序列的顺序，从序列中去掉任意的元素而获得的新序列。例如，字符串acdfg同akdfc的最长公共子串为df，而它们的最长公共子序列是adf。

2.1 最长公共子串算法

最长公共子串算法[5]是计算机科学领域被广泛研究和应用的一类经典算法，也是文本处理的基本算法之一。最长公共子串提取问题可以定义为：给定两个字符串str1和str2，找出str1和str2的所有公共子串中长度最大的子串[6]。

目前，对于最长公共子串的深入研究主要是针对生物基因序列，或是一些由独立字符组成的字符串间的公共子串的获取[5,16,17]。

最长公共子串算法的优点是不需对文本内容进行语言学的预处理，具有语种无关性，适用于中文文本的处理，即可以同时处理中英文文本或者中文文本下的繁简体文本，并且对于提取出的公共子串的长度没有限制。其缺点在于时间复杂度和空间复杂度很大，其空间复杂度会随着文本中字符数量的增加呈平方倍增长，对内存的要求很高。

2.2 N-Gram算法

N-Gram算法可以应用于现代的自然语言处理，即此方法可用于中文文本的处理。其基本思想是将文本内容按字节流进行大小为N的滑动窗口操作，形成长度为N的字节片断序列，每个字节片断称为gram，对全部gram的出现频度进行统计，并按照事先设定的阈值进行过滤，形成关键gram列表，即为该文本内容的特征向量空间，列表中的每一种gram均为一个特征向量维度[7]。

N-Gram算法的优点在于它的语种无关性，可以同时处理中英文、繁简体文本；不需对文本内容进行语言学处理，也

不需词典和规则;对拼写错误的容错能力强。但同时存在很大的缺点,即它限制了提取出的字节片段的长度,可能将原本相连的词或词组分开,使其成为无实际意义的字符串。例如,如果选择 $N=2$,无法提取出长度大于 2 的关键词或词组,如"购物篮相似度"这个词组分割成为"购物"、"篮相"和"似度"这 3 个词,则失去了原本的意义,而"购物篮相似度"的确极有可能是此段文本的主题,对于了解这篇文本具有重要意义,N-Gram 算法由于限制了提取的字节片段的长度,使得提取出的字节片段失去了原本的意义。

目前,有一些基于 N-Gram 算法的深入研究旨在提高分词的准确性[15,18,26]。其中 IKAnalyzer(IK)是一种比较好的针对中文文本的分词技术[15]。与 N-Gram 相比,它不依靠字符数分词,而是根据给定的字符,例如:"也"、"了"、"又"、"的"等字符,当扫描文本时遇到事先定义的字符时进行分词。此方法分词比 N-Gram 算法分词更准确。但是利用此方法进行关键词提取时,需要对分词进行筛选,否则,它的分词几乎相当于整篇文本,准确率会因为词组基数大而降低。它适用于专业文本,例如学术性质的文本和记录性质的文本,对于存在较多音译词和网络新词的文本会出现较多"误分词"的情况。

2.3 Hirschberg 算法

Hirschberg 算法是指 Hirschberg 针对最长公共子串算法发明的改进的线性空间的最长公共子串算法,其基本思想是是使用线性空间来解决最长公共子串问题,主要是为了解决长字符串,该算法使用了动态规划以及分治算法,在时间复杂度上有所增加,但是减少了空间复杂度[8]。

2.4 Nakatsu 算法

Nakatsu 算法是指 Nakatsu 针对最长公共子串问题作出改进的最长公共子串提取算法,简称 Nakatsu 算法。Nakatsu 算法在计算匹配字符串的情况下有着良好的时间复杂度 $O(N(M-P))$和空间复杂度 $O(N^2)$[9]。

Nakatsu 发明的改进的最长公共子串算法与传统的最长公共子串算法相比,在保持时间复杂度和空间复杂度不变的基础上,既具备了传统最长公共子串算法的功能,又可以显示公共子串在字符串中的位置,便于解决单词共现的问题。将此算法应用于文本文档的关键词提取和解决单词共现问题,具有一定的实用价值。它的缺点是算法复杂,且只能应用于英文字母字符串中公共子串的提取,而不适用于有意义的中文文本的公共子串提取,出错率很高。

2.5 关键词提取算法分类

常见的关键词提取算法可归纳为 3 类:基于统计模型的算法、基于语义的算法和基于机器学习的算法[19]。

基于统计模型的关键词提取算法中常见的有词频和 TF-IDF 算法。该类方法简单,不需要训练数据,具备较好的通用性,但准确率较低,一般作为关键词提取算法的基础。

基于语义的关键词提取算法中常见的有互信息和 WordNet 算法。该类方法提高了提取准确率,但是依赖于背景知识库、词典、词表等,对文本格式要求严格,难以推广。

基于机器学习的关键词提取算法是通过对训练样本的训练获得统计参数,构建模型,再将模型运用到关键词提取中,即将关键词提取过程转换成分类过程[20],例如决策树[21]、贝叶斯[22,23]等算法。常见的有基于朴素贝叶斯模型的 KEA (Keyphrase Extraction Algorithm)算法[24]和基于 C4.5 分类算法的 GenEx 关键词提取系统[25]。该类方法提高了关键词提取的准确率,易于推广,但是依赖于选取的算法模型,训练耗时长。

2.6 现有算法和本文算法综合分析

传统的最长公共子串算法虽然算法思想简单,且具有语种无关性,但是其空间复杂度会随着文本中字符数量的增加呈平方倍增长,对内存的要求很高。N-Gram 算法是一种分词算法,且具有语种无关性,但是限制了提取出的字节片段的长度,即可能将原本相连的词或词组分开,使其成为无实际意义的字符串,例如,如果选择 $N=2$,则无法提取出长度大于 2 的关键词或词组,如"文本相似度"这个词组分割成为"文本"、"相似"和"度"这 3 个词,使得其代表的范围很广,并无太大实际意义,而"文本相似度"这个词组对于了解这篇文本具有重要意义。Nakatsu 算法可以提取出两个字符串中不连续的公共子串,并且能得到公共子串中的所有字符在字符串中的下标位置,但是其思想复杂,步骤繁琐,并且只适用于字符串中的字符之间相互独立的情况,如 DNA 序列,这样的字符串不适用于中文文本的处理,如果将此算法应用于中文文本字符串的公共子串提取,有可能提取出的是无意义或断裂的汉字组合。

基于语义的关键词提取算法[27]依赖于背景知识库、词典、词表等,因此对具有较多网络新词和音译词的文本进行关键词提取时,无法提取出不包含于知识库的词或词组。基于机器学习的关键词提取算法依赖选取的算法模型,训练耗时长,更适合于长文本中关键词的提取,不适合短文本。基于统计模型的关键词提取算法思想简单,具有通用性,不需要训练样本,也不依赖于知识库,适用于短文本关键词的提取,因此本文基于统计模型的关键词提取算法,再结合词或词组的长度来进行关键词提取,提高了准确率,无需训练样本,构建模型,可以提取出不包含于知识库中的词或词组。

针对以上算法的优缺点,本文提出了中文文本下的基于公共子串长度和在文本中出现频数的关键词或词组提取算法。该算法的基本思想是:首先提取出中文文本中出现频数大于或等于 2 的词或者词组,并且这些词或者词组的长度也大于或等于 2,然后根据提取出的词或者词组的长度和在文本中出现的频数来衡量权重,依据权重大小筛选出此段中文文本的关键词或词组。

本文所提算法的创新之处在于此方法可以处理包含很多音译词的翻译类中文文本以及包含很多新词的网络上的中文文本,不受词典和语法的限制。与传统的公共子串提取算法相比,其空间复杂度虽然可能会随着字符串长度的增加呈线性倍数增加,但是大部分情况下是不变的,因此对内存的需求与传统的最长公共子串算法相比要小很多。在中文文本中,越长的公共子串越能够准确描述此段中文文本的主旨,因此要尽可能提取出较长的公共子串,这些公共子串长度不一,并且事先并不知道,N-Gram 算法事先规定了分词的长度,且长度统一,这样会将原本应当相连的词组变为离散的词,甚至会失去原本的中文含义,而本文算法对于提取的词或词组没有长度上的限制,不会造成原本有意义的词或词组断裂进而成为没有意义的字符片段。与 Nakatsu 算法相比,Nakatsu 算

法不能对中文文本进行公共子串提取，此算法吸收了 Nakatsu 算法一个的优点，即可以提取出两个字符串中不连续的公共子串，即发现共现的单词或者词组，并且得到公共子串中字符在字符串中的下标。

Wan X. J. 等针对同一篇长文本的摘要和关键词提取会相互影响，提出了基于网络图的长文本摘要和关键词提取算法[14]。本文算法着眼于短文本中关键词提取。

3 基于词或词组长度和频数的中文文本的关键词提取算法

在中文的信息检索和自然语言处理领域中，文本的关键词提取算法占据着十分重要的地位。如何快速准确地找到任意一段中文文本的关键词或词组，从而了解这段文本所要表述的思想，是文本关键词提取算法的重点和难点，也是本文研究的重点。

由于中国历史悠久，因此针对中文的关键词或词组的提取不仅要适用于白话文还要适用于古文。而且随着全球化的发展，不同语言群体间接触增加，中文文本中音译词语增加，再加上网络的普及和网络新词不断涌现，使得中文文本下的关键词或词组提取算法必须具备语种无关性，提取词或词组长度不受限制，不受已有词典约束，同时也要控制提取关键词或词组的时间和空间复杂度。

本文提出了针对中文文本的基于词或词组长度和频数的关键词提取算法，它具有传统的最长公共子串算法的优点，例如语种无关性，对提取的词或词组长度没有限制，能够提取出长度大于或等于 2 的词或词组。同时，本文算法与传统的最长公共子串算法相比，减小了算法的时间复杂度和空间复杂度。本文算法也具有 Nakatsu 算法的优点，它可以得到提取出的词或词组在原文本中的字符下标，以便深入研究单词共现等问题。本文算法考虑到了词或词组的长度和频数方面的因素，词或词组的长度是指词或词组所包含的中文字符的个数，频数是指词或词组在此段文本中出现的次数，体现了词或词组与此段文本主题内容之间的联系。理论分析和实验结果证明，本文提出的基于词或词组长度和频数的中文文本的关键词提取算法在对中文文本进行关键词提取时，与已有的算法相比，在准确率上具有明显的优势。

3.1 算法的思路

在一段文本中，能够代表此段文本的关键词往往会多次出现。在中文文本中，通常具有两个字或两个字以上的词或短语具有较为确实的意义，同时词或词组包含的汉字数目越多，说明这个词或词组包含的信息也越多，越能够准确指明这段文本的主旨。因此，本算法选择关键词或短语出现的频数，以及关键词或词组本身所包含的汉字的数目，作为计算此词权重的重要参数。

3.2 算法

3.2.1 算法的基本思想

整个关键词提取过程为：首先对中文文本进行预处理并分句，使用组合的方法每次取出两个不同的分句，对这两个分句调用本文算法，提取出两个分句中长度大于或等于 2 的公共词或词组，累计存储这些词或词组，从而得到此段文本中频数大于或等于 2 且长度也大于或等于 2 的所有词或词组。最后，计算筛选得到的词或词组的频数，根据这些词或词组的长度和频数计算它们的权重，根据权重得到关键词或词组。

提取两个句子中公共的、长度大于或等于 2 的词或词组的过程为：首先，计算两个中文分句中出现次数最多的汉字字符的出现次数以及较短的分句的字符数；然后创建二维数组，初始化数组中的项为－1；再将两个分句进行对比，将频数大于或等于 2 的词或词组在较长分句中的字符下标记入二维数组中，覆盖初始值－1，如果数组中此行的项不是初始值，即初始值已经被前面的公共字符下标覆盖过，则覆盖下一行同一列的初始值；最后得到两个句子中公共的、长度大于或等于 2 的词或词组。此步骤的时间复杂度为 $O(n^2)$。

本文算法主要应用于网络上包含新词的短文本和由外国文献翻译来的包含众多音译词的中文短文本，对它们分句，分句数目设为 n，对分句进行两两排列组合的时间复杂度为 $O(n^2)$；由于这些文本长度较小，包含的分句的数目 n 也较小，因此在实际处理文本时能够在较短的时间内完成处理。

3.2.2 算法的流程图

包括预处理、本文算法执行和后处理在内的整体流程如图 1 所示。

算法 1 Keyword Extraction Algorithm based on Length and Frequency of words or phrases for Short Chinese texts (KEALENF)

输入：一段中文文本

输出：中文文本中根据公共子串的频数和长度取得的关键词或词组

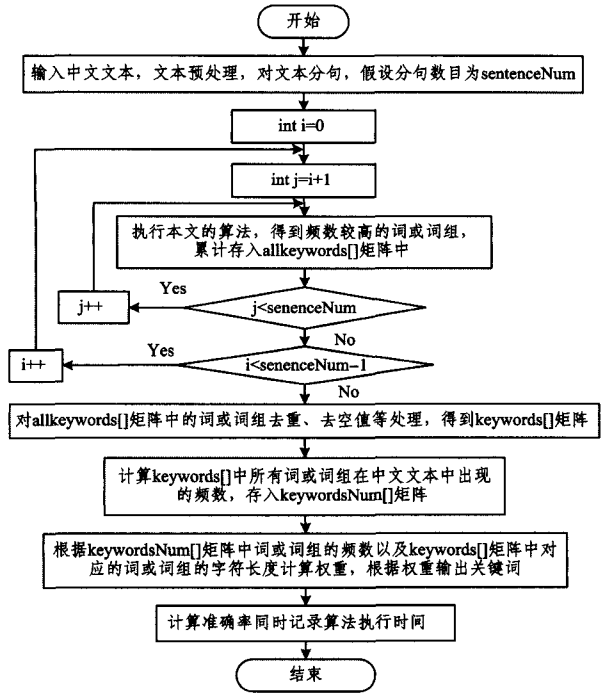

图 1 整体流程图

注意：1）$keywordsNum[i]$ 表示 $keywords[i]$ 在文本中出现的频数；2）本文中 $keywords[]$ 矩阵中词或词组的权重计算方法为：$keywordsNum[i] * keywords[i].length$。

提取频数较大的词或词组的算法流程如图 2 所示。

算法 2 提取两个句子中公共的、长度大于或等于 2 的词或词组的算法

输入：两个文本分句 str1 和 str2

输出：两个分句中的公共子串

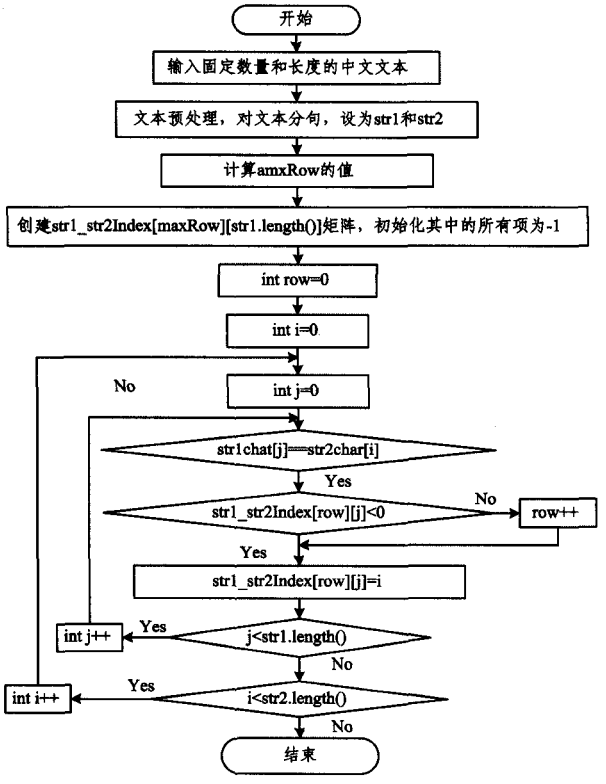

图 2　频数较高词或词组提取流程图

注意：1）假设 *str*1 的字符数小于 *str*2 的字符数；2）*maxRow* 表示 *str*1 和 *str*2 两句话中出现次数最多的汉字出现的次数；3）*str*1_*str*2*Index*[*maxRow*][*str*1. *length*()]矩阵用于存储公共子串字符下标；4）*row* 表示 *str*1_*str*2*Index*[*maxRow*][*str*1. *length*()]矩阵的列值变换，*i* 表示 *str*2 中字符下标，*j* 表示 *str*1 中的字符下标。

3.2.3　算法的伪代码

输入：固定数量和长度的中文文本

输出：中文文本中根据公共子串的频数和长度取得的关键词或词组

步骤：

1. 对文本进行简单的预处理，去掉此段文本中除了“，”和“。”以外的其他标点符号，包括回车(\n)、水平制表符(\t)、空格(\s)和换行(\r)等，并根据句号分句。假设给定分句矩阵中任意两个中文句子，设为 str1 和 str2，将其转成字符矩阵 str1char[]和 str2char[]存储；
2. 定义一个整型的变量 maxRow，用来记录两句话中出现次数最多的汉字出现的次数；
3. 创建存储公共子串字符下标的矩阵，这里假设 str1 的字符长度小于 str2 的字符长度，存储公共子串字符下标的矩阵记为：str1_str2Index[maxRow][str1. length()]，初始化其中的所有项为－1；

4 设置一个整型变量 row=0；

5. for(str2 中的每个字符：i){
6. for(str1 中的每个字符：j){
7. if(str2 中的某个字符与 str1 中的某个字符相同){
8. if(存储公共子串字符下标的矩阵没有被修改过，仍为初始值－1，即 str1_str2Index[row][j]==－1){
9. 置存储公共子串字符下标的矩阵中对应位置的项的值为 str2 中与 str1 中字符相同的字符的下标，即：str1_str2Index[row][j]=i；
10. }else if(存储公共子串字符下标的矩阵已经被修改过，不等于初始值－1，即 str1_str2Index[row][j]>－1){
11. 表示此行上这个字符已经对比过且确认在此字符前出现过，已经将原本初始化的值－1 替换为 str2 中对应的字符下标，因此要在下一行对应列的位置将－1 替换为 str2 中此字符对应的下标：row＋＋；str1_str2Index[row][j]=i；
12. }end if
13. }end if
14. }end for
15. }end for
16. 使用 for 循环扫描存储公共子串字符下标的矩阵(假设 str1 的长度小于 str2 的长度)，str1 _ str2Index[maxRow][str1. length()]，将其中连续两个及以上项大于－1 的项分别提取出来，即为提取出的所有公共子串的字符下标；根据提取出的公共子串的字符下标，到 str2 中找到对应的字符串，即为公共子串，每次都将提取到的公共子串存储到用于存储公共子串的数组 allkeywords[]中，最终得到这段中文文本中频数大于或等于 2 的词或词组；
17. 对提取出的 allkeywords[]中的词或词组去重，即将重复的词或词组去掉，得到唯一出现的词或词组，并将其存储到 keywords[]矩阵中；
18. 计算 keywords[]中所有词或词组在中文文本中出现的频数，用 keywordsNum[]矩阵记录；这里 keywordsNum[i]表示 keywords[]中下标 i 的字符串 keywords[i]在文本中出现的频数，keywords[]矩阵和 keywordsNum[]矩阵是一一对应的；
19. 根据 keywordsNum[]矩阵中词或词组的频数以及 keywords[]矩阵中对应的词或词组的字符长度计算权重；里计算方法为：keywordsNum[i] * keywords[i]. length，得到 keywords[i]的权重，并根据权重取出关键词；
20. 计算准确率，同时记录算法的执行时间。

3.3　算法详述

3.3.1　预处理

创建表格，记录中文文本和其存储位置的映射，以方便在测试算法时能根据表格中的映射关系逐个调出文本；对中文文本进行预处理，只需要将此段文本中除了“，”和“。”以外的其他标点符号如回车(\n)、水平制表符(\t)、空格(\s)和换行(\r)等删除，并根据句号分句，对分句两两调用本文提出的算法。

3.3.2　算法详细步骤

假设分句后任意取两个句子记为 sen1 和 sen2，将其转成字符序列，计算这两个句子中出现次数最多的汉字的频数，记为 *h*，*h* 即为记录重复出现的字符下标的二维数组的行数。将这两个字符序列中长度较小的字符序列的长度作为二维数组的列数，记为 *r*，并初始化二维数组中的项为－1；然后将较长的字符序列的字符与较短的字符序列的字符逐个比对，如果较长的字符序列中的一个字符和其下一个字符与较短的字符序列中连续两个字符均相同，才将二维数组中对应位置的项设为较长的字符序列中字符的下标，否则不变，从而在二维表中只记录了存在两个及两个以上字符连续出现的字符下标；接着，通过二维数组中出现的连续的字符下标，得到两个句子的字符串，并将其存入指定的用于存储关键词或词组的数组中；反复执行上面的步骤，直至提取出所有的公共子串。

3.3.3　后处理

将存储于关键词或词组的数组中的元素做汇总处理，即汇总其在文本中的频数，同时根据每个字符串的长度和其在文本中出现的频数计算其权重，按照权重的大小进行降序排

序。根据文本的长度，取出前 n 个公共子序列，即为此段文本的关键词或词组。

3.4 算法分析

本文算法的优点在于：1)算法思想简单，通过简化词或词组提取的操作步骤减小了时间复杂度，减小了过渡矩阵的大小及时间复杂度和空间复杂度；2)对提取的词或词组没有内容和长度上的限制，可以更准确地指代文本的主旨。提取的词或词组在内容上没有限制是指算法具有语种无关性，可以处理文言文和包含少量英文或者音译词组的中文文本，对音译的词或词组的容错力强；提取的词或词组在长度上没有限制是指提取到的关键词或词组的长度不一，但尽量获取完整的子串，不会出现将“数据挖掘”这个词分成“数据”和“挖掘”这两个词的情况，从而使得对文本内容的分析出现偏差。

综上所述，此算法尤其适用于现今网络上海量的中文文本信息关键词提取，以及由外国文献翻译而来的中文文本。网络上时时有新词出现，不论词典如何更新，这些新词极有可能不包含于参考词典中。由外国文献翻译而来的中文文本中包含很多的音译词，包括一些外国人的名字、地区的名称或者一些中文无法意译出且具有特定的文化背景的外语单词或词组，例如“Las Vegas”“拉斯维加斯”、“Sri Lanka”“斯里兰卡”、“Chanel”“香奈儿”、“Louis Vuitton”“路易威登”等。利用此算法可以将文本中这些新词和音译的词准确地提取出来，从而便于快速准确地了解文本的内容。

此算法的缺点在于：此方法可以提取出在文本中出现频数大于或等于 2 并且长度也大于或等于 2 的词或词组，不能提取出只在文本中出现一次的单个中文字符。

在中文文本中，与主题密切相关的词或词组只出现一次的情况是很少见的。因为，在中文文本中为了明确主题，与主题相关的词或词组一般会多次重复出现。反之，如果将出现一次的词或词组也提取出来，提取出的词或词组的数目会非常多，筛选是一个很大的问题。如果文本中有“例如”、“包括”等词时，后面将有很多的专业词或词组，则更加难以解决这些词的筛选问题。因此，无法提取出频数小于 2 的词或词组这点不能完全算是缺点，它降低了后期对提取出的词或词组筛选的难度。

在中文文本中，只用一个汉字字符来描述主题的情况几乎没有。因为，单个的汉字字符可以引申出很多的词或词组，因此其没有明确的指代意义。例如，“纸”可以引申为“餐巾纸”、“草稿纸”、“报纸”、“造纸术”等。因此，即便无法取出单个中文字符，其一般也不会对关键词的提取有太大的影响。

4 实验分析

本文从包括 Google 学术、百度学术和万方数据知识平台搜索到的中文论文的摘要和关键词、微博文本及一些由外国文献翻译过来的中文文本中采集了实验用到的文本数据。

测试方法：对中文论文的摘要分别采用传统的最长公共子串算法、N-Gram 算法、IKAnalyzer[15] 算法、Nakatsu 算法和本文提出的基于词或词组长度和频数的中文文本的关键词提取算法进行关键词提取，并分别记录关键词提取所需时间，通过将提取的关键词与论文作者在论文中指定的摘要对应的关键词对比，得到准确率。至于微博文本和一些由外国文献翻译过来的中文文本，也分别采用传统的最长公共子串算法、N-Gram 算法、IKAnalyzer[15] 算法、Nakatsu 算法和基于词或词组长度和频数的中文文本关键词提取算法进行关键词提取，并分别记录算法执行时间，将提取的关键词与人工提取关键词进行对比，计算算法提取关键词的准确率，并利用折线图对比准确率和算法执行时间。

4.1 实验环境

实验数据集采用 txt 文本格式的 1000 篇中文文本，分别利用传统的公共子串提取算法、N-Gram 算法、Nakatsu 算法以及本文提出的基于词或词组长度和频数的中文文本的关键词提取算法，按照文本长度以及固定的测试数目进行测试。提取这些中文文本的关键词，并将其与文本的主题词进行对比，计算其准确率和关键词提取所需时间，对比算法提取关键词的准确率和效率。

实验采用 Windows XP 操作系统，Intel Core2 Duo 2.9 GHz CPU，2GB RAM；编程工具为 MyEclipse，使用 MySQL 数据库，采用 Java 编写算法相应代码进行测试。

4.2 算法实验及分析

算法的准确率只与测试文本的长度相关，与测试文本的数量无关。测试文本长度越大，提取的关键词也越多，经过权重计算后得到的关键词也就越准确，与文本主题相关度越大，即准确率越高。但是准确率与测试文本的数量无关，因为算法每次都是针对测试文本逐个测试，文本数量的多少不会影响算法的准确率。

执行算法所需的时间与测试文本的长度和测试文本的数量密切相关。测试文本的长度越长，执行算法所需时间也越长；同样，文本数量越多，执行算法所需的时间也越多。

本文算法的空间复杂度明显小于 Nakatsu 算法和传统的公共子串提取算法的。由于执行算法过程中中间矩阵的创建方式不同，Nakatsu 算法和传统的公共子串提取算法的中间矩阵会随着文本分句长度的增长呈平方倍增长，空间复杂度也随着文本分句长度的增长呈平方倍增长。本文算法的中间矩阵虽然会随着文本分句长度的增长而增长，但是增长的幅度最多只是线性增长，如果只是较长分句的长度增长，其中间矩阵大小不变。因此，空间复杂度要明显小于其他算法。

4.2.1 关键词提取准确率实验分析

本文算法着重于提高中文文本提取关键词的准确率，关键词提取的准确率与测试文本的长度密切相关，如图 3 所示，横坐标表示测试的中文文本的字符数，纵坐标表示提取出关键词的准确率。该方法首先筛选提取出的词或词组，接着将筛选后的词或词组的字符数乘以在文本中出现的频数，再除以所有提取出的公共子串各自的字符数与各个子串在文本中出现的频数的乘积。计算公式为：准确率＝筛选后的词或词组的字符数 * 在文本中出现的频数/(所有提取出的词或词组的字符数 * 其在文本中出现的频数)。

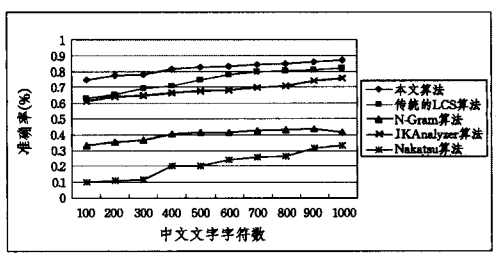

图 3 提取关键词的准确率与文本长度的关系

本文算法的准确率之所以高于传统的最长子串算法、N-Gram算法和Nakatsu算法，是因为它考虑到了词或词组在文本中出现的频数及不限制词或词组的长度，因此提取出的词或词组不会像N-Gram算法那样被“截断”，导致提取出来的词或词组失去其原本的含义。又因为本算法考虑到了词或词组在文本中出现的频数，所以其与文本的主题更为贴近。IKAnalyzer算法虽然分词较为准确，但是只进行了分词，本身并不具有关键词提取的功能。综上，此算法提取关键词的准确率高于传统的最长子串算法、N-Gram算法和Nakatsu算法。

4.2.2 执行算法所需的时间实验分析

执行算法所需的时间会随着测试文本长度的增加而增加，如图4所示；也会随着测试文本数目的增加而增加，如图5所示。

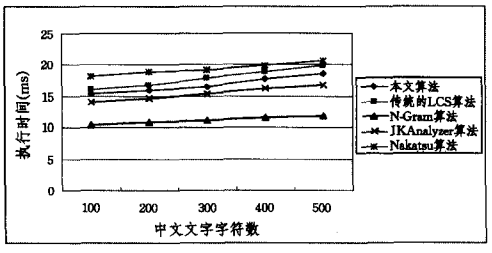

图4 提取关键词所需时间与文本字符数的关系

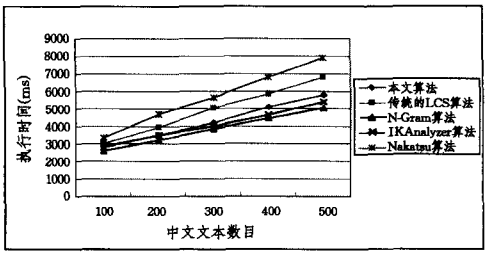

图5 提取关键词所需时间与文本数量的关系

图4中，横坐标表示用于测试的中文文本的字符数，纵坐标表示执行相应的算法测试单个文本所需的时间，单位是毫秒(ms)。由于针对单个文本，算法执行时间太短，多次针对单个测试文本执行得到的时间不一，因此选择了50～100个长度较为一致的文本来测试算法的执行时间，并求取平均值，得到图4所示的结果。

本文算法在进行词或词组提取时，需对文本的分句进行公共子串提取，因此其时间复杂度与传统的公共子串提取算法最为接近，但是本文算法在进行频数较高的词或词组提取时做了改进，有效地降低了其空间复杂度，同时降低了扫描临时数组所需的时间，因此本算法时间复杂度略小于传统的公共子串提取算法。Nakatsu算法在进行频数较高的词或词组提取时，步骤更为复杂，因此时间复杂度最高。而N-Gram算法只是一个分词过程，因此时间复杂度最低。IKAnalyzer算法要按照给定的字符进行分词，其中有一个匹配过程，因此耗时会较多。

图5中，横坐标表示用于测试的中文文本的数目，纵坐标表示针对相应横坐标数量的测试文本执行算法所需的时间，单位是毫秒(ms)。由于算法的执行时间与应用于测试的中文文本长度相关，因此此处选用的测试文本的长度在[450，550]区间内，得到图5所示的结果。

本文算法在进行词或词组提取时，要先对文本的分句进行两两组合，再对两个分句进行词或词组的提取，因此执行时间会随着文本数目的增加而增加。但是本文在进行分句时，对分句的长度和数量进行了处理，将仅包含3个汉字字符以下的分句与其后面的分句合并。例如，一个分句只包含3个字符，则将其与其后面的分句合并成为一个分句，从而使得分句的数目尽量少，因此，复杂度只是呈线性增长。

综合以上的实验结果分析，本文的算法适用于中文文本的关键词提取，准确率高于传统的公共子串提取算法、N-Gram算以及Nakatsu算法；本文算法的执行时间在测试文本长度或者测试文本数量相同的情况下比Nakatsu算法稍长，但是小于传统的公共子串提取算法和N-Gram算法。

结束语 关键词提取技术是文本信息处理的重点和难点，大部分关键词提取算法都是针对英文文本操作的；针对中文文本的关键词提取算法很少且准确率不高，本文提出了基于词或词组长度和频数的中文文本的关键词提取算法。

面对各类中文文本，包括不断有新词出现的网络文本以及由外国文献翻译过来的含有很多音译词的中文文本，本文研究了如何借助计算机快速、准确地提取出文本中的关键词或词组从而了解这些中文文本主题的算法。

此算法最主要的优势在于它着眼于中文文本中的频数较高、字符数较多的词或词组的提取，尤其适用于现今网络上海量的中文文本以及由外国文献翻译过来的中文文本的关键词提取。网络上时常有新词出现，这些新词极有可能不包含于已有的词典中，外国文献翻译过来的中文文本中包含很多的音译词，新词和音译词均为关键词提取算法的难点。本文算法可以将文本中的这些新词和音译的词准确地提取出来，便于快速、准确地了解文本的内容。对于提取的关键词或短语，此算法不限制提取的词或词组的长度；提取的词或词组是连续的汉字字符，这些词或词组的频数越大、包含汉字字符数越多，此词或词组权重越高；此算法在提取词或词组时取出了关键词在句中的起始点和终止点下标，基于此可以深入探讨单词共现方面的问题。

就准确率而言，由上面的实验可知，本文算法在提取关键词和词组时准确率明显高于其他算法。就时间复杂度而言，此算法与传统的最长公共子串(LCS)算法相比，稍有改进；但与Nakatsu算法相比，本文算法的时间复杂度更高。就空间复杂度而言，此算法具有明显优势。此算法在查找公共子串时的过渡矩阵的行数很小，不仅小于传统的最长公共子串算法的过渡矩阵，也小于Nakatsu发明的改进的最长公共子串算法的过渡矩阵。

而且，本文算法经过如下扩展可以用于英文文本中的关键词提取：如果将每个空格分开的一个英文单词看作是一个汉字字符，可以使用本文中提取词或词组的算法提取出英文文本中的词或词组，并筛选出关键词。因此，本文算法也可以用来提取英文文本中的关键词或词组。综上所述，本文算法不仅可以用于中文文本下的关键词提取或者中文文本中含有其他语言的混合文本下的关键词提取，也适用于英文文本下的关键词提取，此算法具有完全的语言无关性特点。

参考文献

[1] Manaris B. Natural language processing: A Human-Computer

Interaction Perpective [R]. Computer Science Department. University of Southwestern Louisiana Lafayette: Advanced in Computers. Louisiana Volume 47, 1999: 1-66

[2] Wang Hui, Zhang Wei-de, Zeng Qiang, et al. Extracting important information from Chinese Operation Notes with natural language processing methods[J]. Journal of Biomedical Informatics, 2014, 48(2014): 130-136

[3] Che Hai-yan, Fen Tie, Zhang Jia-chen, et al. Automatic Knowledge Extraction from Chinese Natural Language Documents[J]. Journal of Computer Research and Development, 2013, 50(4): 834-842(in Chinese)
车海燕,冯铁,张家晨,等. 面向中文自然语言文档的自动知识抽取方法[J]. 计算机研究与发展,2013,50(4):834-842

[4] Zong Cheng-qing. Statistical Natural Language Processing[M]. Beijing: Tsinghua University Press, 2013: 5(in Chinese)
宗成庆. 统计自然语言处理[M]. 北京:清华大学出版社,2013:5

[5] Iliopoulos C S, Rahman M S. New efficient algorithms for the LCS and constrained LCS problems[J]. Information Processing Letters, 2008, 106(2008): 13-18

[6] Pan Hong, Xu Chao-jun. Application of LCS-Based Algorithm in Chinese Term Extraction[J]. Journal of the China Society for Scientific and Technical Infomation, 2010, 29(5): 853-857(in Chinese)
潘虹,徐朝军. LCS算法在术语抽取中的应用研究[J]. 情报学报,2010,29(5):853-857

[7] Sidorov G, Velasquez F, Stamatatos E, et al. Syntactic n-grams as machine learning features for natural language processing[J]. Expert Systems with Applications, 2014, 41(3): 853-860

[8] Hirschberg D S. A Linear Space Algorithm for Computing Maximal Common Subsequences[J]. Communication of the ACM, 1975, 18(18): 341-343

[9] Nakatsu N, Kambayashi Y, Yajima S. A longest common subsequence algorithm suitable for similar text strings[J]. Acta Informatica, 1982, 18(2): 171-179

[10] Sproat R, Shih C. A statistical method for finding word boundaries in Chinese text[J]. Computer Processing of Chinese and Oritental Languages, 1990, 4: 336-351

[11] Liu Zhi-yuan, Chen Xin-xiong, Sun Mao-song. Mining the interests of Chinese microbloggers via keyword extraction[J]. Frontiers of Computer Science in China (FCSC), 2012, 6(1): 76-87

[12] He Gan-jun. Multimensional Investigation of Chinese Transliterration Words[J]. Journal of Jiangxi Social Sciences, 2012, 32(4): 194-197(in Chinese)
何干俊. 汉语音译词的多维考察[J]. 江西社会科学,2012,32(4):194-197

[13] Lv Shu-xiang, Ding Sheng-shu. Modern Chinese Dictionary[M]. Beijing: Dictionary of the Language Institute of the Chinese Academy of Social Sciences; Foreign Language Teching and Research Press, 2003: 2148-2149(in Chinese)
吕叔湘,丁声树. 现代汉语词典[M]. 北京,中国社会科学院语言研究所词典编辑室:外语教学与研究出版社,2003:2148-2149

[14] Wan Xiao-jun, Yang Jian-wu, Xiao Jian-guo. Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction[C]// ACL 2007. 2007

[15] Chen H. Research on Chinese segmentation algorithm based on Hadoop cloud platform[C]// 2015 Information Technology and Mechatronics Engineering Conference. Atlantis Press, 2015

[16] Chen Chuan-peng, Qin Zhong-ping. A Systolic Architecture with Linear Space Complexity for Longest Common Subsequence Problem[C]// IEEE 8th International Conference on ASIC, 2009 (ASICON'09). 2009: 33-36

[17] Ye Ning, Zhu Da-ming, Zhang Qian-qian, et al. A Fast Algorithm of Constrained Longest Common Subsequence[J]. Journal of Nanjing University(Natural Sciences), 2009, 45(5): 576-584 (in Chinese)
业宁,朱大铭,张倩倩,等. 带约束最长公共子序列快速算法[J]. 南京大学学报(自然科学版),2009,45(5):576-584

[18] Zhai Zhong-wu, Xu Hua, Li Jun, et al. Sentiment Classification for Chinese Reviews Based on Key Substring Features[C]// International Conference on Natural Language Processing and Knowledge Engineering, 2009(NLP-KE 2009). IEEE, 2009: 1-8

[19] Han Xue-jiao. The Research of Keyword Extraction Algorithms on English Short Test Text[D]. Beijing: North China University of Technology, 2013(in Chinese)
韩雪娇. 英语试题关键词抽取算法研究[D]. 北京:北方工业大学,2013

[20] Wang Bing-kun, Huang Yong-feng, Yang Wan-xia, et al. Short text classification based on strong feature thesaurus [J]. Journal of Zhejiang University Science C(Computers & Electronics), 2012, 13(9): 649-659

[21] Zhang Yun-tao, Gong Ling, Wang Yong-cheng. An improved TF-IDF approach for text classification[J]. Journal of Zhejiang University Science A(Science in Engineering), 2005(1): 50-56

[22] Zhang Feng, Fan Xiao-zhong, Xu Yun. Chinese Term Extraction Based on PAT Tree[J]. Journal of Beijing Institute of Technology(English Edition), 2006(2): 162-166

[23] Bu Tao, Wang Ji-cheng, Huang Yuan. Design and Implementation of Chinese Document Automatic Classification System[J]. Journal of Chinese Information Processing, 1999, 13(3): 26-32 (in Chinese)
部涛,王继成,黄源. 中文文档自动分类系统的设计与实现[J]. 中文信息学报,1999,13(3):26-32

[24] Chen Ping, Zhou Chang-le, Lian Rui-ting. An Improved Approach to Keyword Extraction Using KEA[J]. Journal of Mind and Computation, 2011(2): 48-54(in Chinese)
陈平,周昌乐,练睿婷. 一种改进的 KEA 关键词抽取算法研究[J]. 心智与计算,2011(2):48-54

[25] Zhang Liang, Zou Fu-tai, Ma Fan-yuan. KRBKSS: a keyword relationship based keyword-set search system for peer-to-peer networks[J]. Journal of Zhejiang University Science A(Science in Engineering), 2005(6): 577-582

[26] Feng Yong, Li Hua, Zhong Jiang, et al. Text Classification Algorithm Based on Adaptive Chinese Word Segmentation and Proximal SVM[J]. Computer Science, 2010, 37(1): 251-254(in Chinese)
冯勇,李华,钟将,等. 基于自适应中文分词和近似 SVM 的文本分类算法[J]. 计算机科学,2010,37(1):251-254

[27] Fang Jun, Guo Xiao, Wang Xiao-dong. Semantically Improved Automatic Keyphrase Extraction[J]. Computer Science, 2008, 35(6): 148-151(in Chinese)
方俊,郭霄,王晓东. 基于语义的关键词提取算法[J]. 计算机科学,2008,35(6):148-151