知识抽取中的嵌套向量分词技术

化柏林 赵 亮

（中国科学技术信息研究所 北京 100038）

【摘要】 向量分词算法已经比较成熟，通过在知识抽取过程中实现向量分词算法，对向量切分中的关键技术进行归纳总结，同时发现一趟向量切分的不足，并针对这些不足，设计嵌套的向量分词技术。实验证明，在知识抽取过程中，采用嵌套的向量切分方法，不但切分准确率高、切分全面，而且能从根本上解决“词中有词”的问题，有利于后续的句法分析。

【关键词】 知识抽取 最大向量法 词法分析 分词技术 嵌套向量分词 【分类号】 TP391 G356

Nested Vector Segmentation Technique in Knowledge Extraction

Hua Bolin Zhao Liang

(*Institute of Scientific and Technical Information of China*, *Beijing 100038*, *China*)

【**Abstract**】 Well - known algorithm of maximum matching method is implemented in the process of knowledge extraction, and drawn a conclusion about critical techniques of vector segmentation. Nested vector segmentation is designed and implemented on account of disadvantage of once scanning. According to experiment, nested vector segmentation is used in knowledge extraction, it not only improves precision and recall, which resolves the problem of word in word radically, but also provides convenience to following syntactic analysis.

【**Keywords**】 Knowledge extraction Maximum matching method Lexical analysis Segmenting technique Nested vector segmentation

1 引 言

向量切分按长度分为最大向量法与最小向量法，按方向又分为正向向量切分、逆向向量切分和双向向量切分，这样组合起来一共有 6 种向量切分方法，分别为正向最大向量法（也称最大匹配法、MM 法、the Maximum Matching Method）[1]、正向最小向量法、逆向最大向量法（也称逆向最大匹配法，OMM、RMM、IMM 法[1-4]）、逆向最小向量法、双向最大向量法与双向最小向量法。最大匹配法与逆向最大匹配法是向量切分中的典型代表，但最大匹配法或最小匹配法等概念都不能涵盖以上 6 种方法，因此称为向量匹配法或机械分词法更为合适[2]。对于现代汉语，最小匹配是不适用的，因为书面汉语中的每一个字几乎都可以成为一个词[2]。

向量分词的基本算法已比较成熟，而根据不同的应用需求，许多学者在此基础上对最大匹配法进行了改进。文献[5]设计了“正向扫描 + 增字最大匹配（包括跳跃匹配）+ 词尾歧义检查 + 归右原则（对连续型交集，需左部结合）”的改进算法。文献[6]在分词词典上采用区间最大词长，改进正向减字最大匹配法为“词首 + 长词匹配 + 短词推进”自动标引方法。文献[7]为把数据库的自然语言查询转换成 SQL 语句，提出了比较适用于数据库查询的递归式最大匹配法。文献[8-10]在现有最大匹配法的基础上，针对“长词优先”原则来对其进行改进，使得每次匹配都是在整个句子范围内优先寻找最长词，从而解决切分歧义的问题。

本文在已有的向量分词算法基础上，针对知识抽取中后续句法分析的需要，设计了一种嵌套向量分词方法，该方法除考虑准确率和速度两个指标外，还着重考虑切分的颗粒度，判断已切分的正确结果是否还能再分，无论是长词本身还是长词中所含的短词都需要切分出来。笔者首先用 VBA 在 Excel 里进行模块实验，然后选取合适的算法用 Java 进行系统实现，包括向量切分基本算法、词典的组织与查找技术、嵌套的向量分词技术等。

收稿日期：2007 - 05 - 11
收修改稿日期：2007 - 05 - 22

2 向量切分关键技术

在用向量切分法进行分词时,首先把词典装入内存并排序,然后读入待切分文件,每完成一句切分,就把结果写到目标文件里。以正向最大向量切分法为例,首先读入待分析串,如果没有到句子末尾,就取向量长度个字符,然后去词典里匹配,如果找到,就作切分处理;如果没找到,就去掉最右字符,再到词典里查找;如果只剩一个字符,就作单字处理。切分处理包括 3 点:把该词切分出来并加上切分标记,指针向后移动(移动词长个字符),最大向量初始化。单字处理也包括 3 点:把单字切分出来不加切分标记,指针向后移动一个字符,最大向量初始化。正向最大向量切分流程如图 1 所示,其算法实现如例 1 所示:

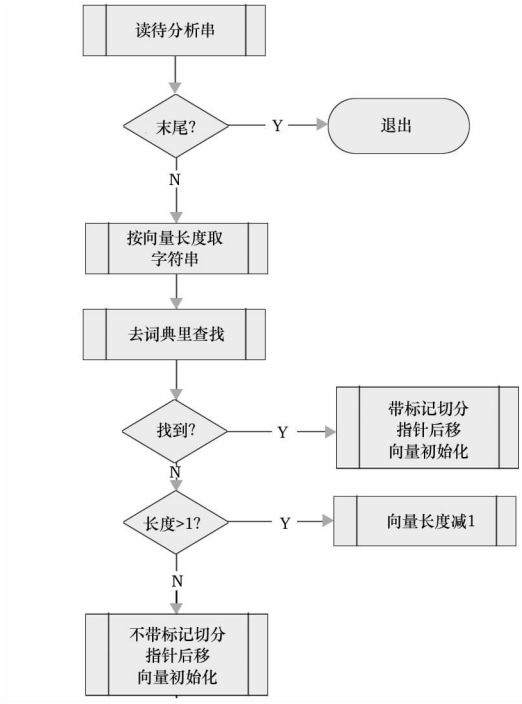

图 1 正向最大向量切分的程序流程图

例 1:正向最大向量法的算法一核心代码(segMaxVector 方法)

```
iVector = iMaxVector;
for (i = 0;i < = sSentence. length( ) ;i ++ ) {
  sWord = sSentence. substring(i,iVector) ;
  int iLocation = Arrays. binarySearch(arrayLexicon,sWord) ;
  //在词典里找到,就做切分处理
  if (iLocation > = 0) {
    sSegment = sSegment + " " + sWord + "/ ";
    i = i + iVector - 1;
    iVector = iMaxVector;
  } else {
    //如果没有找到就去掉最右字符
    if(iVector > 1 ) {
      iVector = iVector - 1;
      i = i - 1;
    //双字符都找不到,那就是单字了
    } else {
      sSegment = sSegment + sSentence. substring(i,1) ;
      iVector = iMaxVector;
    }
  }
}
```

上述程序中,实际上有 3 个分枝:正常词串切分、单字切分、不切分。正常切分的条件是在字典里找到匹配的词,处理结果是把该词切分出来,加斜杠空格作以标记;单字切分的条件是 iVector 已经变得很小,小到只有一个字符,处理结果是把它切出来,不加空格也不加标记,一方面,可能是停用词;另一方面,可能是多个单字构成未登录词。不切分的条件是词典里没有找到该词而且长度足够长,处理结果是向量长度变短。

向量切分的核心是切分过程中词长向同一个方向变化,称之为“向量”,最大向量法是保证长词优先。在具体实现时,最大词长的确定难度很大,如果设定的太短,那么长词就切分不出来;如果设定的太长,那么每次切分,都会有很多无谓的匹配。

3 向量切分的词典排序与查找技术

为了提高向量切分的速度,就要减少查找与匹配次数,而减少查找与匹配次数的关键就是改善词典的组织方式与查找算法。向量切分法所依赖的词典可以按词条长短分组,并按使用频率排序。在实际切分过程中,首先把词典读到内存里,并根据词长分别存放到不同的数组里,这样,词典的长度就会小很多,遍历起来就会大大减少匹配次数。如果采用顺序遍历,那么词典就应当按频率降序排列;如果采用二分查找[11],那么词典就应当按字符编码进行排序。无论查找采取哪种匹配方法,词典采取哪种排序方式,按词条长短进行分组并事先读到内存是个不错的选择。表 1 显示了一个句子在最大向量设为 5 的情况下详细切分过程。

上述过程中,用 25 步过程切出 5 个有效词(带斜杠与空格)和 3 个单字(只带空格)。每一步过程都要到词典里查找一遍。切分一个句子到词典的查找次数为 $\sum_{i=1}^{n}(\mathrm{iMaxVector}-\mathrm{Len}(W_i)+1)$,也就是执行这么多次的词典查找,查找次数等于最初设定的向量长度与句中每个词长的差的累加和。设句子长度为 iLenSen,平均词长

为 iLenWord，初始向量长度为 iMaxVector，句子切分结果的词数为 iWord。当 iMaxVector > iLenSen/iWord + iLenWord 时，查找次数等于句子长度；当 iMaxVector = iLenSen/iWord + iLenWord 时，查找次数等于句子长度；当 iMaxVector < iLenSen/iWord + iLenWord 时，查找次数小于句子长度。对于最终能切分出来的词，它的查找次数为 $iMaxVector - len(W_i) + 1$；对于最终未能切分出来的每个字符，它所需要的查找次数为 iMaxVector - 1。

表 1 “讨论了信息伦理学与互联网的关系”的切分过程

	待切分串(下划线部分表示当前取字串)	切分结果
1	文章讨论了信息伦理学与互联网的关系	
2	文章讨论了信息伦理学与互联网的关系	
3	文章讨论了信息伦理学与互联网的关系	
4	文章讨论了信息伦理学与互联网的关系	
5	讨论了信息伦理学与互联网的关系	文章/
6	讨论了信息伦理学与互联网的关系	文章/
7	讨论了信息伦理学与互联网的关系	文章/
8	讨论了信息伦理学与互联网的关系	文章/
9	了信息伦理学与互联网的关系	文章/ 讨论/
10	了信息伦理学与互联网的关系	文章/ 讨论/
11	了信息伦理学与互联网的关系	文章/ 讨论/
12	了信息伦理学与互联网的关系	文章/ 讨论/
13	了信息伦理学与互联网的关系	文章/ 讨论/
14	信息伦理学与互联网的关系	文章/ 讨论/ 了
15	与互联网的关系	文章/ 讨论/ 了 信息伦理学/
16	与互联网的关系	文章/ 讨论/ 了 信息伦理学/
17	与互联网的关系	文章/ 讨论/ 了 信息伦理学/
18	与互联网的关系	文章/ 讨论/ 了 信息伦理学/
19	互联网的关系	文章/ 讨论/ 了 信息伦理学/ 与
20	互联网的关系	文章/ 讨论/ 了 信息伦理学/ 与
21	互联网的关系	文章/ 讨论/ 了 信息伦理学/ 与
22	的关系	文章/ 讨论/ 了 信息伦理学/ 与互联网/
23	的关系	文章/ 讨论/ 了 信息伦理学/ 与互联网/
24	关系	文章/ 讨论/ 了 信息伦理学/ 与互联网/ 的
25		文章/ 讨论/ 了 信息伦理学/ 与互联网/ 的 关系/

每次查找所用的花费(即去词典里比较的次数)依赖于该词所在的位置，如果位置靠前，所用花费就比较少；如果位置靠后，所用的花费就比较多；如果该串不在词典中，那么就要遍历整个词典一遍。切分一个句子所需要的比较次数为 $P_i \times \sum_{i=1}^{n}(iMaxVector - len(W_i) + 1)$，公式中 P_i 为每个词在词典中的位置。实验中的词典有 43 980 条，按照二八原则，每个词的位置平均在 8 796，不进行分组的情况下，采取顺序遍历，整个句子要到词典里执行 8 796 ×25次，也就是 219 900 次比较。这个计算量还是相当大的，因此词典一定要分组，而且要排序，还要读到内存里。

4 嵌套向量切分技术

单纯的使用一趟向量分词技术不能解决“词中有词”的问题。如果使用最小向量切分，那么长词就分不出来；如果使用最大向量切分，那么长词中所包含的短词就分不出来。嵌套向量切分技术就是对初次切分结果中长度较长的已切分串进行嵌套切分。如“知识管理”中含“知识”与“管理”，它们各自成词，使用嵌套向量切分不仅可以切出“知识管理”，还可以切出“知识”与“管理”。

嵌套切分有两种方式：一种是对已切分串用最小向量进行二次切分；另一种是对已切分串递归调用最大向量分词程序进行切分。对已切分串的判断是从串中的最后一个空格开始到“/” 标记结束，即从标记符开始向前取，取到第一个空格为止。如对“所 适合 的 管理范式/”里的长串进行嵌套切分，识别的标记为从离“/”最近的空格到“/”，这之间的内容为二次切分的潜在对象，如果它的长度比较长，就应当执行二次切分。用最小向量进行二次切分的程序如例 2 所示：

例 2：最大向量中嵌套最小向量切分程序

```
For i = 1 To Len(sSource)
  iFind = InStr(1, sSource, "/ ")
  If iFind > 0 Then
    sTemp = Trim(Mid(sSource, 1, iFind - 1))
    iSpace = getLastSpace(sTemp)
    If iSpace > 0 Then
      sTemp = Mid(sTemp, iSpace + 1)
    End If
    sSource = Mid(sSource, iFind + 2)
    '//未切分字符串长度足够长，进行二次切分
    If Len(sTemp) > 2 Then
      sReplace = segMinVector(sTemp, 2)
      secondSegment = Replace(secondSegment, sTemp, "[" +
        sReplace + "]")
    End If
    i = 1
  End If
Next
```

采用嵌套向量切分技术可以把长词中的短词分出来，同时保留对长词的索引。在知识抽取系统中，经过自动分词后就进行词性标记。词性标记过程中所有的短词都需要标记，词典中没有收录的长词一般不作明确的词性标记。而词性标记完成后就进行句法分析，在句法分

析过程中,嵌套向量切分技术可以很容易地把短词归约成长词,同时把未进行词性标记的长词标记成短语。

实验中,整个分词过程采取了多趟扫描的处理方式。首先,用关键词词表把关键词切分出来,然后,用概率词典把长度较长的未切分串进行切分,如"、所适合的"切分成"、所 适合 的",最后把长度较长的关键词利用概率词典作二次切分,如"图书馆联盟"可以切分为"图书馆/ 联盟/"。表 2 显示了对一篇文摘的多趟处理结果。

表 2 某篇文摘多趟处理结果

第一趟切分结果	文章/ 从 知识管理/ 与 博客/ 的 发展历史/ 着手,从 管理方式/ 、管理理念/ 、所 提供/ 的 知识内容/ 、所适合的 管理范式/ 、产生/ 的 影响/ 等方面 分析/ 了两者的 差异/ ,并提出 知识/ 博客/ 在 图书馆联盟/ 中的重要 应用/ 。
第二趟切分结果	文章/ 从 知识管理/ 与 博客/ 的 发展历史/ 着手 , 从 管理方式/ 、管理理念/ 、所 提供/ 的 知识内容/ 、所 适合 的 管理范式/ 、产生/ 的 影响/ 等 方面 分析/ 了 两者 的 差异/ , 并 提出 知识/ 博客/ 在 图书馆联盟/ 中 的 重要 应用/ 。
嵌套切分结果	文章/ 从[知识/ 管理/] 与 博客/ 的[发展/ 历史/]着手 , 从[管理/方式/]、[管理/ 理念/]、所 提供/ 的[知识 /内容/]、所 适合 的[管理/ 范式/]、产生/ 的 影响/ 等 方面 分析/ 了 两者 的 差异/ , 并 提出 知识/ 博客/ 在[图书馆/ 联盟/]中 的 重要 应用/ 。
词性标记结果	文章/n 从/p [知识/n 管理/n]/NP 与/p 博客/n 的/u [发展/n 历史/n]/NP 着手/v ,/w 从/p [管理/n 方式/n]/NP ,/w [管理/n 理念/n]/NP ,/w 所/u 提供/v 的/u [知识/n 内容/n]/NP ,/w 所/u 适合/v 的/u [管理/n 范式/n]/NP ,/w 产生/v 的/u 影响/vn 等/u 方面/n 分析/v 了/u 两者/r 的/u 差异/n ,/w 并/c 提出/v 知识/n 博客/n 在/p [图书馆/n 联盟/n]/NP 中/f 的/u 重要/a 应用/vn 。/w

5 结 语

实验证明,向量切分的关键在于:待切分串的截取技术、词典的组织技术以及词典的匹配技术。待切分串的截取技术涉及截取初始位置与截取长度的变化;词典的组织技术主要涉及索引方式与排序方式的选择;词典的匹配技术主要使用等值比较,也就是精确比较,所以词典的查找算法便成了匹配的关键。

知识抽取不同于信息检索,知识抽取不仅需要分词,还要进行句法分析、语义分析甚至语用分析,才能把大量文献中所蕴含的知识点抽取出来,存入知识库。进行嵌套分词有利于句法分析时的归约处理,如根据分词结果,可以很容易地把"知识/n "与"管理/n "归约成"知识管理/NP"。

在自然语言处理整个过程中,分词是处理的基础与关键,但并不是自然语言处理的瓶颈。在测试中,逆向最大向量分词的准确率已达 99.6%(单纯使用逆向最大匹配的错误率为 1/245[6,8]),这已能满足绝大多数应用的需求。句法分析也已渐渐成熟,其实自然语言处理的瓶颈应该是语义的分析与理解。制约语义分析发展的因素主要有两个:一个是知识库的匮乏,让计算机拥有可以利用的知识,特别是常识知识太困难,尽管本体研究的新热给知识库的构建带来了一定的帮助,但本体的构建与自动获取仍然有相当大的难度,与知识获取一样是人工智能的瓶颈。另一个制约因素是语义表示比较困难,找不到一个通用的无二义性的覆盖面宽的语义表示方式,无论是逻辑语言还是语义网,都无法完整地描述整个复杂的语言系统。自然语言处理的应用非常广泛,但要取得突破性进展依然有很大的困难,只能循序渐进地不断建设资源库与改进算法,从而提高处理的准确率。

参考文献:

[1] 梁南元. 书面汉语的自动分词与一个自动分词系统—CDWS[J]. 北京航空学院学报,1984,(4):97 - 104.

[2] 揭春雨,刘源,梁南元. 论汉语自动分词方法[J]. 中文信息学报,1989,3(1):1 - 9.

[3] 关英春,秦蓓. 汉语文字自动统计系统[J]. 中文信息学报,1986,(1):26 - 32.

[4] 揭春雨,刘源,梁南元. 汉语自动分词实用系统 CASS 的设计和实现[J]. 中文信息学报,1991,5(4):27 - 34.

[5] 骆正清,陈增武,胡上序. 一种改进的 MM 分词方法的算法设计[J]. 中文信息学报,1996,10(3):30 - 37.

[6] 王兰成. 基于 EMM 中文抽词算法的 XMARC 主题信息挖掘[J]. 情报学报,2005,24(1):82 - 86.

[7] 赵元正,戴尔晗. 基于递归式最大匹配法的数据库查询接口的实现[J]. 计算机时代,2006(12):38 - 40.

[8] 苏芳仲,林世平. Web 文本挖掘中的一种中文分词算法研究及其实现[J]. 福州大学学报(自然科学版),2004,32(增刊):67 - 71.

[9] 路永刚,赵伟. 一种改进的 MM 分词方法的研究与实现[J]. 长春工业大学学报(自然科学版),2006,27(4):320 - 323.

[10] 郑逢斌,付征叶,乔保军,等. HENU 汉语自动分词系统中歧义字段消除算法[J]. 河南大学学报(自然科学版),2004,34(4):49 - 52.

[11] 马玉春,宋瀚涛. Web 中文文本分词技术研究[J]. 计算机应用,2004,24(4):134 - 136.

(作者 E - mail:huabolin@ istic. ac. cn)