北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis

藏文数词识别与翻译

孙萌[1, 2, †]　华却才让[3]　刘凯[1]　吕雅娟[1]　刘群[1]

1.中国科学院计算技术研究所，北京　100190; 2. 中国科学院研究生院，北京　100049;
3. 青海师范大学藏文信息研究中心，西宁　810008; † E-mail: sunmeng@ict.ac.cn

摘要　通过对藏文数词内部构词规律及外部边界信息进行分析，提出对藏文数词基本构件定义的方案。采取最优路径决策模型判断数词构件边界，然后通过有限自动机模型识别并翻译基本数词，最后用模板匹配算法处理复杂数词。结果表明，提出的方法对数词识别与翻译的 F 值达到 98.73%，在藏汉机器翻译的测试集上的 BLEU 提高了 2.64。

关键词　藏文；数词基本构件；自动机；数词识别；数词翻译

中图分类号　TP391

Tibetan Number Identification and Translation

SUN Meng[1,2, †], HUA Quecairang[3], LIU Kai[1], LÜ Yajuan[1] ,LIU Qun[1]

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190; 2. Graduate University, Chinese Academy of Sciences, Beijing 100049; 3. Tibetan Information Research Center, QingHai Normal University, Xining 810008;
† E-mail: sunmeng@ict.ac.cn

Abstract　Tibetan number identification and translation is essential for Tibetan-Chinese translation. The authors propose a definition of Tibetan number basic component through analyzing the inner structure and the boundary information. A best path decision was applied in judging basic component, then the number was recognized and translated by a finite automation model, finally a template matching algorithm was used for processing complicated number. The *F*-score of identification and translation is 98.73% and the BLEU score of Tibetan-Chinese translation obtains an improvement of 2.64.

Key words　Tibetan; number basic component; automation; number indentification; number translation

藏文是一种具有逻辑格语法体系的拼音文字，经过1400多年不断的发展，已经成为我国藏族地区和其他藏传佛教传播地广泛使用的文字，同时藏文也是世界上最复杂的语言文字之一。藏文的复杂性表现在两个方面。一方面，藏文属于汉藏语系藏缅语族，词之间缺少明显标记。另一方面藏文是拼音文字，由 30 个辅音字母、4 个元音字母以及上、下加字组成，并且藏文的拼写形式又可以分为横向和纵向拼写，在时态上具有曲折变化。藏文分词[1-5]已经是藏文信息处理[6]中的一个难题，而藏文数词的构成方式更加灵活多样，普通的统计模型已不适用于藏文数词识别与翻译。

在藏文中，数词通常有 3 种表达方式。第一种是阿拉伯数字，比如“2012”；第二种是藏文基本数字构件：༠(0), ༡(1), ༢(2), ༣(3), ༤(4), ༥(5), ༦(6), ༧(7), ༨(8), ༩(9)，比如“༢༠༡༢(2012)”；第三种是藏文组合数词，组合数词是由藏文数词基本构件以一定的规则组合而成，比如“སུམ་བཅུ་ལྔ(35)”。前两种表达形式通过简单的匹配和映射就可以实现识别和翻译。但是藏文组合数词的构成规律复杂多变，并且藏文的数词基本构件通常具有歧义性，因为数词的基本构件也会作为普通词的组成部分。

863 计划(2011AA01A207)资助
收稿日期: 2012–06–05；修回日期: 2012–08–24；网络出版时间：2012-10-26 17:49
网络出版地址：http://www.cnki.net/kcms/detail/11.2442.N.20121026.1749.011.html

藏文的数词识别与翻译模块是藏汉翻译系统中不可或缺的组成部分，然而国内外研究者对其研究很少，仅有 Liu 等[7]提出的基于数词组件分类的藏文数词识别算法。Liu 等第一次系统地阐述了藏文复杂的数词构成方式，并总结归纳为 12 条基本规则。预先定义 5 种数词构件的集合，相应定义 6 种标签，分别是基本数词、前缀数词、连接数词、后缀数词、独立数词和非数词。通过查询音节所属的集合初始化序列的标签，经过若干迭代操作修改标签，最终识别数词。然而，由于藏文本身丰富的表达方式和灵活的构词规律，所谓的数词构件可能是一个非数词的组件，从而导致识别的错误。其次，在论文中提到的 5 种数词构件集合存在重合现象，应该考虑左右邻信息避免分类错误。另外，藏文数词有多种构词方式表示同一个数值，藏文数词的翻译也是一个难点。最后，Liu 等提出的多次迭代修改标注的方法，时间复杂度较高。

本文中，我们深入研究了藏文数词的构词规律，对藏文数词构件进行更为细致的分类，并添加数词消歧模块。采用三层模型，自底向上，准确识别并翻译藏文数词。

1 文数词组成规律

藏文的组合数词譬如类似汉语中的“十五”，“一百二十六”，通常是由基本数词构件按照一定规律组合连接。藏文从一到九可以看作基本构件，如“གཅིག གཉིས་ གསུམ་ བཞི་ ལྔ་ དྲུག་ བདུན་ བརྒྱད་ དགུ་”。单位词百、千、万、十万、百万、千万、亿也是基本构件，如“བརྒྱ་ སྟོང་ ཁྲི་ འབུམ་ ས་ཡ་ ཉི་ཁ་ དུང་ཟློག་”。然而，基本构件会有较多变体，比如，“一”也可以写做“དང་཮”、“ཅིག”和“གཅིག”等。从 1 到 100 藏文的表达方式较为多样，语言现象较多，而某些构词方式更倾向于习惯语法，不遵从语言专家总结的构词规律。因此我们将基本数词构件的范围扩展到 1 到 100，大于 100 的藏文数词的构词方式相比而言更容易用规则描述，构建面向大于 100 的藏文数词的规则库会很大的减少规则的数量，并会极大地避免规则冲突。

目前，国内外没有对藏文数词的种类和范围统一规范的定义，本文综合考虑藏文的数词构词规律和规则存储使用方式，将藏文数词划分为 5 类：基本数词、序数词、分数、单位词和时间词。

基本数词 仅有数词构件组成的数词，如“སུམ་སྟོང་བརྒྱ་དང་བཅུ་གཅིག (3111) 和ཁྲི་བཅུ་གཅིག (11 万)”。

序数词 表示次序的数词，如“ཕྱོགས་གསུམ་པ་ (第三届)。”

分数 包括普通分数，如“སུམ་ཆའི་གཉིས་(三分之二)，百分数、千分数和万分数，如བརྒྱ་ཆའི་གཉིས་(百分之二)”。

单位词 单位+基本数词，如“སྦྱི་ལེ་ཀྲ་བཞི་མ་ཉི་ཤུ་ (二十平方公里)”。

时间词 表示时间概念的词，如“༡༩༩༠ལོ་(一九三零年)”。

2 总体框架

由于藏文数词构词特有的复杂性，统计方法不能精确刻画其复杂的规律，而单一的规则模型的描述能力有限，为避免多粒度规则之间的包含和冲突，将提高单一规则模型构建的复杂度。我们采用三层规则模型，将藏文数词识别与翻译的任务划分为 3 个独立的阶段，即边界识别、基本数词识别与翻译和复杂数词识别与翻译，简化问题从而降低模型的复杂度。由于藏文句子中没有分隔符，第一层模型用以识别基本数词构件；第二层模型根据识别出来的基本数词构件，通过有限状态自动机识别并翻译基本数词；最后一层模型把基本数词泛化为变量，识别更大粒度的复杂数词。如图 1 所示。

首先，根据基本构件表将藏文句子切分并构建成一个有向图，通过最优路径算法将正确的数词基本构件识别出来；然后，采用自动状态机表述基本数词规则，仅需扫描一遍句子便可识别并翻译出基本数词；最后，将识别出的基本数词泛化为一个变量，通过复杂数词规则进而识别并翻译出更复杂的

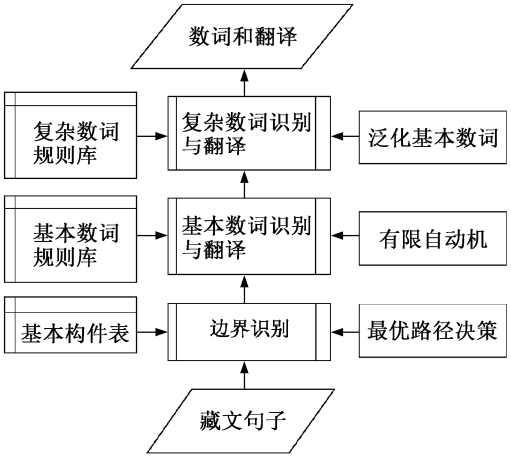

图 1 藏文数词识别与翻译流程图

Fig. 1 Flow diagram of Tibetan number identification and translation

数词, 如日期、序数词和单位词等。

3 层叠式模型

藏文数词识别的过程, 可以划分为状态识别和状态转换两个阶段, 而数词翻译根据状态转移路径生成对应译文。藏文属于拼音文字, 音节可以作为其基本构词单位, 但是音节与音节之间缺少明显分隔符, 因此藏文数词识别的第一步是确定数词构件的边界。藏文数词基本构件可以定义 7 种类别, 即对应 7 种不同输入, 可采用有限自动机描述状态之间的转移。

3.1 边界识别模型

根据藏文数词基本构件的含义和语法功能, 我们将其划分为以下 7 种类别:

基数词 0 到 100 之间的藏文数词及其变体, 据大规模语料统计, 平均每个基数词有一到两个变体。基数词既可以单独表示数词意义, 又能够遵循特定语言规律组合成复合数词。

数量词 单位词指具有百、千、万、十万、百万、千万和亿等含义的词, 藏文单位词为“བརྒྱ་/སྟོང་(སྟོང་ཐུགཔ་)/ཁྲི་/(ཁྲི་ཚོ་ ཁྲི་ཐུགཔ་)/འབུམ་(འབུམ་ཐུགཔ་)/ས་ཡ་/ ཟུ་བ་/ དུང་ཕྱུར་”。藏文单位词的位置较为灵活, 可以放于基数词之前和之后。

数词前缀 数词前缀通常为表示 1 到 9 含义基本数词, 但是其中 1 至 3 有其特殊形式, 其余前缀数词与基数词集合存在交集。数词前缀通常和单位词搭配。

小数点 包含两种形式: ཚེག་/གྱངས་ཀྱང་”。主要用以表示小数。

连接词 不具有数词意义, 但具有语法功能。

否定数词 指མེད་, 在数词中表示“否定”的意思。如 ས་ས་སྟོང་དྲུག་བརྒྱ་བཅུ་མེད་གསུམ་(3603), 在基数词བཅུ་(10)之后的否定数词མེད་, 表明“没有 10”, 即十位数位置上应该设为零。

数词后缀 不影响数词的含义, 仅具有语法功能。如“ར་/བ་” 。

本文定义的 7 类构件, 存在以下 3 个问题: 首先, 数词前缀集合和基数词集合存在重合; 其次, 集合中某个数词构件也可能是另一个数词构件的子串; 最后, 所定义 7 类数词构件在实际使用中还可能被用做普通词的构件。

解决上述 3 个问题, 有两种方案。一是将各种类型的规则都用一个模型建模, 融合不同类型规则会增加模型的复杂度, 并且, 过多规则带来的规程冲突和相互影响也会增大编码实现的难度。二是用多个模型, 每个模型仅处理一类或几类问题, 层层推进, 模型之间接口清晰, 模型边界定义简单, 不仅简化了实现的难度, 更易于今后的维护。

第一层模型主要解决数词构件的边界问题。

藏文数词识别的第一步就是要确定数词构件的边界。边界识别模型主要是对句子进行初步切分, 得到一个数词构件粒度合适的粗分结果。粗分结果的准确性与包容性直接影响后续的两个模块的效果, 并最终影响整个数词识别与翻译系统的正确率和召回率。边界问题包含数词构件本身边界判断和数词与非数词的边界判断。

数词构件本身边界识别存在一定的歧义性。例如, 一个数词构件同样也可能是另一个数词构件的子串。数词 “ས་སྟོང་བརྒྱ་དང་བཅུ་གཅིག (3111)”, 其中“གཅིག(1)”、“བཅུ་(10)”和“བཅུ་གཅིག(11)”均可被识别为基数词, 但在此数词中, 基数词是“བཅུ་གཅིག(11)”, 而“གཅིག(1)”和“བཅུ་(10)”只是“བཅུ་གཅིག(11)”的子串。正向最大匹配和逆向最大匹配算法可以在一定程度上解决数词构件识别的歧义性, 为了提高识别的准确率, 本文采用基于词图的识别算法。

首先, 根据本文定义的数词构件词典, 找出句子中所有可能的数词, 构造数词识别的有向无环图。图中的边表示一个数词构件, 并赋予相应的权重。然后将在此词图中的最优路径作为数词构件的识别结果。如图 2 所示。

求解有向图最优路径问题可以选择时间复杂度为 $O(V^2)$的 Dijkstra 算法, 其中 V 表示图中顶点的数量。但是本文用到的词图属于按照拓扑顺序排列节点的有向图, 因此可以简化 Dijkstra 算法, 从左到右

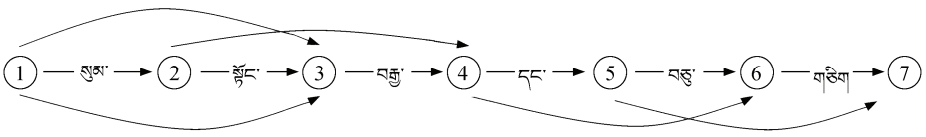

图 2 数词构件识别的词图

Fig.2 Lattice of number basic components

依次计算每个节点与源点之间的最短路径, 时间复杂度为 $O(E)$, E 表示图中边的数量。

如算法 1 所示, G 表示输出的词图, w 表示边的权重, π记录顶点的前驱, Adj[u]表示顶点 u 的邻接边的集合, $d[u]$表示顶点 u 到源端的距离。

数词与非数词的边界初步识别是本层模型的第二个任务。

我们将包含数词构件的普通词语定义为混淆词语。如 “གཅིག་སྒྲིལ་(统一)”中粗体部分还可以表示数词“一”, “སྡེ་ལེ་ཐུ་བཞི་མ་(平方公里)”中粗体部分也可以表示数词“四”。我们可以通过构建一个混淆词表, 如果句子中有混淆词, 则在词图上增加一条混淆边, 混淆边与普通边竞争, 由最优路径算法选择最为合适的切分。

算法 1 生成最短路径

```
SHORTESTPATH(G,w)
for each vertex u, taken in topologically sorted order
   do for each vertex v∈Adj[u]
      do RELAX(u,v,w)
RELAX(u,v,w)
if d[v]>d[u]+w(u,v)
   then d[v] ← d[u]+w(u,v)
        π[v] ← u
```

3.2 基本数词识别与翻译模型

藏文句子中数词构件的边界确定之后, 藏文句子被切分为数词构件和非数词构件组成的序列。基本数词识别与翻译模型是本系统的核心模块, 将上述的序列通过有限自动机识别藏文基本数词并翻译。

通常的字符串识别往往可以借助正则表达式引擎, 只需人工书写正则表达式即可识别出符合要求的字符串。因而正则表达式在传统的文本匹配与替换方面起到非常重要的作用。但是藏文基本数词识别与翻译, 因其问题的特殊性, 不能直接使用正则表达式实现识别与翻译功能。一是因为藏文数词与汉语数词存在表达上的差异, 不能直接通过正则表达式直接替换相应构件进行翻译, 要以阿拉伯数词作为翻译的“中间语言”, 在识别的同时进行数学运算计算生成最终的阿拉伯数词; 二是边界识别模型没有过多考虑基本构件在句中的上下文信息, 又因为上文定义的 7 种构件类型存在部分重合, 还需在本模型中根据上下文信息反馈修改之前确定的构件种类; 三是假设用正则表达式表示所有的数词规律, 并且规则的数量为 M, 输入序列的长度为 N, 则识别此序列中数词的时间复杂度为 $o(M*N)$, 本模型采用自动机算法, 只需扫描一遍序列, 即可识别并翻译出所有的基本数词, 时间复杂度为 $o(N)$。

基本数词的识别与翻译可以用下面的状态转换图(图 3)表示。

基于有限自动机的藏文数词识别与翻译模型可以看做弧上有标记的有向图, 标记的集合就是数词基本构件的种类。将句子以构件作为输入标记, 进行状态转移寻找路径, 一旦找到一条从开始节点 S 到结束节点 E 的路径, 即识别为一个藏文基本数词。并且在状态转移的同时, 自动机内部维持一个变量, 用以记录从开始节点到当前节点的数词的“值”。

例 1 藏文数词ཉེས་བྱེ་དགུ་བརྒྱ་བཅུ་མེད་ བདུན་(20907)

1) 先进行数词基本构件划分, 得到如下序列: ཉེས་(前缀词–二) བྱེ་(数量词–万) དགུ་ (基数词–九) བརྒྱ་(数量词–百) བཅུ་(数量词–十) མེད་(否定词–没有) བདུན་(基数词–七)。

2) 依次扫描输入的构件序列, 进行状态转移: S→1→2→3→2→2→5→3→E。

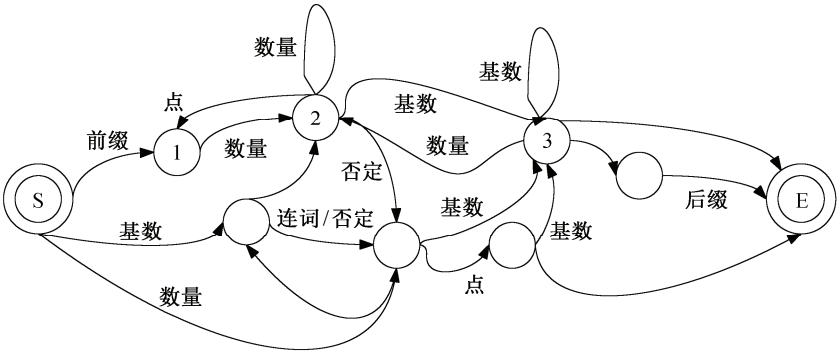

图 3 基本数词状态转换图

Fig. 3 Basic number state switch

例 2 数词བདུན་དང་གངས་ཆུང་གཅིག་ལྔ་(7.15)。

1) 划分基本构件: བདུན་(基数词-七) དང་(连词–和) གངས་ཆུང་(点) གཅིག་(基数词–一) ལྔ་(基数词–五)

2) 状态转移: S→4→5→3→3→E。

藏文数量词的位置较为灵活, 可以是“基数词/前缀词+数量词”或者“数量词+基数词/前缀词”, 在翻译中就面临数量词是修饰前面的词还是后面的词的问题, 可以归结为藏文数量词的前向后向属性判别问题。仅回溯查看前一个输入构件的种类, 不足以确定数量词的前向后向属性, 需再次回溯, 从而根据前两个输入构件的种类, 判别数量词的前向后向属性。如 ཁྲི་ཚ་(数量词-万) སུམ་(前缀词-三) སྟོང་(数量词-千), 首先数量词“千”和“三”组合, 然后再和数量词“万”结合, 译文应该是“三千万”。

3.3 复杂数词的识别与翻译模型

基本数词并不包括时间词、序数词、分数和单位词等复杂数词。可以通过人工书写识别翻译模板处理复杂数词。将基本数词泛化为变量 *X*, 如果在句子中匹配上对应的模板, 则识别并翻译复杂数词, 否则仅输出基本数词。

根据规则中变量 X 的数量, 可以划分为一元规则, 二元规则, 三元规则。为避免规则的冲突, 应该先从三元规则往下进行匹配。譬如, “*X*1 年”和“*X*1 年 *X*2 月 *X*3 日”, 应该遵循最大匹配原则, 首先尝试匹配较大模板, 如不可匹配则再匹配较小模板。

复杂数词识别与翻译的规则定义如下。

序数词 རྒྱལ་*X*→第 *X* 届

时间词 *X*1ལོའི་ཟླ་*X*2ཚེས་*X*3 →*X*1 年 *X*2 月 *X*3 日

分数 *X*1ཆ་*X*2→百分之 *X*2 (当 *X*1=100)

单位词 སྨིག་*X*1→*X*1 米

复杂数词识别与翻译的策略是: 首先将基本数词泛化为变量 *X*, 按照规则表中的优先级依次用规则匹配, 如果匹配上, 则将原句子中的匹配部分标注为已处理, 否则, 尝试下一条规则。

4 实验和分析

4.1 数词识别与翻译实验

我们从政府法律文献和藏文网页中随机抽取 2117 句含有藏文数词的句子, 由人工标注和翻译之后作为测试集。从识别和翻译两方面考察系统的性能。实验结果如表 1 所示。识别是指仅考察识别数词的精度, 而识别与翻译指考察识别与翻译同时正确的精度。

表 1 藏文数词识别与翻译实验结果

Table1 Result of Tibetan number identification and translation

评测指标	识别	识别与翻译
准确率	0.9857	0.9845
召回率	0.9920	0.9908
F 值	0.9888	0.9873

从实验结果可以看出识别的 *F* 值比识别与翻译的 *F* 值稍高, 这是因为数词翻译存在一对多的现象, 比如 “༡༩༠༣ལོ་” 可以翻译成“一九三零年”或“一九三〇年”, 藏文词“ལོ”, 可以表示“岁”或者“年”的含义, 因此必须根据具体语境判断, 单纯依靠规则不能解决。另外, 对于表示概数意思的藏文数词, 还不能很好地处理, 譬如 “སྟོང་གསུམ་བཞི་ཆ་ཤས་གཅིག་” (三四千一份)。本系统可以准确处理规范的藏文数词, 对于倾向于口语化的藏文数词的识别并翻译, 准确率依然达到了可以接受的标准。

4.2 数词识别与翻译对藏汉翻译的影响

藏文数词识别与翻译是藏汉翻译中虽小但必须要解决的问题。因为藏文数词特有的灵活性, 不能通过单纯的统计模型处理, 本文的主要工作就是研究如何表示存储和使用专家知识, 从而构建高效高质的藏文数词识别与翻译系统。将数词识别与翻译模块融入机器翻译模型以提升翻译质量, 基本有两种方案: 1) 在翻译前将对应的数词译文替换藏文数词, 然后整句进行翻译; 2) 在翻译前将藏文数词以及其译文作为短语对动态加入到翻译模型中。方案一会在一定程度上影响翻译模型的完整性, 本文选择方案二作为系统融合的策略。

在本实验中, 以汉语作为目标端。采用 CWMT2011 提供的藏汉双语句对作为训练集语料, 表 2 列出藏汉双语训练集的统计信息。使用 CWMT2011 提供的开发集作为本次试验的开发集, 共 650 句。我们从训练集中预留 604 句含有藏文数词的句子作为测试集。采用 GIGA 新华语料上训练的 5 元语言模型, 平滑方法采用 Kneser-Ney smoothing。

我们使用实验室内部开发的层次短语解码器[8]作为基线系统, 层次短语解码器可以使用从双语语料中抽取的翻译规则, 是近年来主流的机器翻译技术。将藏文数词识别与翻译作为句子翻译前的预处理模块, 即在翻译每句之前将数词以及其翻译动态的加入到基线系统的规则表中。实验结果如表 3 所

表 2　训练语料库统计信息
Table 2　Statistics of training corpus

	藏文	汉语
句子	101629	101629
词语	1280787	971520

表 3　翻译性能对比
Table 3　Result of translation

系统	开发集	测试集
层次短语基线系统	0.4263	29.13
添加藏文数词识别与翻译	0.4338	31.77

示。

从实验结果可以看出，引入数词识别与翻译模块之后，在测试集上 BLEU 提高 2.64 个点。性能提高的原因在于减少了由于藏文数词带来的未登录词。

本文提出的层叠式的模型，将藏文数词识别与翻译任务划分为边界识别、基本数词识别翻译和复合数词识别翻译 3 个相对独立的模块，提高识别与翻译的精度的同时，还降低单一规则模型构建的复杂度。在 2117 句实际网络文本的测试集上，取到 0.9873 的 F 值，验证论文提出方法的鲁棒性。在翻译实验中，引入数词识别与翻译模块，提升翻译质量，表明模型具有很强的实用性。

5　结语

藏文的数词识别与翻译是一项比较基础，但很重要的工作，但是国内外的相关研究较少。本文根据藏文数词的构词规律，定义藏文数词类别和基本构件。提出三层模型，通过基于最优路径决策的数词构件边界识别模型、基于自动机的基本数词识别与翻译模型和基于泛化变量模板的复合数词识别与翻译模型，使得藏文数词识别与翻译的 F 值达到 98.73%。将此模块加入到翻译模型中，翻译效果也得到提升。在现有研究的基础上，我们将进一步研究如何利用上下文信息以更好地指导数词识别。

参考文献

[1] 陈玉忠，李保利，俞士汶. 藏文自动分词系统的设计与实现. 中文信息学报, 2003,17(3): 15－20

[2] 才智杰. 藏文自动分词系统中紧缩词的识别. 中文信息学报, 2009, 23(1): 35-37

[3] 孙媛，罗桑强巴，杨锐，等. 藏语交集型歧义字段切分方法研究 // 第十二届中国少数民族语言文字信息处理学术研讨会论文集. 新疆, 2009

[4] 刘汇丹，诺明花，赵维纳，等. SegT: 一个实用的藏文分词系统. 中文信息学报, 2012, 26(1):97－103

[5] Liu Huidan, Nuo Minghua, Ma Longlong, et al. Tibetan Word segmentation as syllable tagging using conditional random fields // Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation. Singapore , 2011:168–177

[6] 高定国，关白. 回顾藏文信息处理技术的发展. 西藏大学学报: 社会科学版, 2009, 24(3): 18－27

[7] Liu Huidan, Zhao Weina, Nuo Minghua, et al. Tibetan number identification based on classification of number components in Tibetan word segmentation // Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, 2010: 719–724

[8] Chiang D. Hierarchical phrase-based translation. Computational Linguistics, 2007, 33: 201–228