北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis
doi: 10.13209/j.0479-8023.2014.001

藏文文本自动校对方法及系统设计

珠杰[1,2,†]　李天瑞[1]

1. 西南交通大学信息科学与技术学院，成都 610031；
2. 西藏大学计算机科学系，藏文信息技术研究中心，拉萨 850000；† E-mail: trocky.jie@gmail.com

摘要　以藏文音节拼写检查、梵音转写藏文检查、接续关系检查、词语检查为研究内容，提出藏文文本自动校对框架和接续关系检查算法。根据该框架及算法，设计并实现藏文自动校对系统。通过实验证明算法和系统的可靠性和有效性。

关键词　藏文音节；自动校对；接续关系

中图分类号　TP391

An Approach for Tibetan Text Automatic Proofreading and Its System Design

ZHU Jie[1,2,†], LI Tianrui[1]

1. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031;
2. Department of Computer Science, Tibetan University, Lhasa 850000; † E-mail: trocky.jie@gmail.com

Abstract　Taking Tibetan syllable spelling check, Sanskrit transliteration in Tibetan check, connective relation and words check as the objects, a framework of Tibetan text automatic proofreading and an algorithm of Tibetan connective relation are proposed. Under the framework and algorithm, a Tibetan text automatic proofreading system is designed and implemented. Reliability and effectiveness of the algorithm and system are confirmed through corresponding experiments.

Key words　Tibetan syllable; automatic proofreading; connective relation

文本自动校对是一项复杂的自然语言处理过程，包括拼写检查、真词错误检查、语法检查、自动纠错等内容，是自然语言处理的基础工作。从目前的研究现状来看，藏文自动校对方法的研究文献还不多，如多杰卓玛[1]以线性化的藏文音节为研究对象，提出利用三元模型的藏文音节的校对方法，该模型丢失了藏文纵向拼写的特征，也没有校对效果进行实验验证。刘文香[2]提出藏文音节规则来校对藏文音节设想，但没有具体的模型，也没有相应的校对算法。才让卓玛等[3]利用藏文音节规则和分词方法，提出音节和词语校对的方案，区分音节、词语和句子校对 3 种不同的类型。这些文献对藏文文本的自动校对进行了初步讨论，但没有深入的研究藏文自动校对的特殊性，既没有考虑错误的不同种类、藏文接续关系的特殊性，也没有进行比较充分的实验验证。本文首先分析藏文文本中 5 种可能出现的错误：藏文音节拼写错误、梵音转写藏文词语错误、词语错误、接续关系错误和语法错误，在此基础上，针对前 4 种错误类型，提出不同的错误识别方法，并通过实验验证方法的有效性，再进一步设计自动校对系统来验证藏文自动校对框架的可行性。由于语法错误的复杂性，本文暂不进行探讨。

国家自然基金项目(61262058)、CCF 中文信息技术开放基金项目(CCF2012-02-01)和藏文信息技术教育部“长江学者与创新团队发展计划”(IRT0975)资助

收稿日期：2013–05–30；修回日期：2013–08–23；网络出版时间：2013-11-11 10:25

1 藏文文本自动校对系统

藏文文本自动校对系统是一个复杂的系统，包括藏文音节拼写检查、梵音转写藏文检查、藏文接续关系检查、词语校对、语法语义检查等内容，贯穿自然语言处理领域的字处理、词法分析、句法分析、语义分析的内容。下面从藏文文本错误类型、校对系统框架设计和系统实现思路等方面进行描述。

1.1 藏文文本错误类型

英文文本校对中，常见的错误类型有非词错误、真词错误和句法语义错误。针对藏文的情况，本文定义了如下 5 种类型的错误。

定义 1 藏文音节拼写错误是指不符合藏文字性组织规则的无效藏文音节。 例如“གཅིག”写成“གཅིག”，“སྒྲུབ་”写成“སྒྲུབ་”等。这些错误可能是由于人为的输入错误，或者正字法知识的缺陷，造成的拼写错误。

定义 2 梵音转写藏文错误是指由音节点隔开的藏文字符串不符合梵音转写藏文文法规则的无效梵音转写藏文。例如“ཀཆ་བཀའ་བཞུད”写成“ཀཆ་བཀའ་བཞུད”等。

定义 3 接续关系错误是指不符合藏文格助词、不自由虚词接续关系文法的连接错误。例如“རྒྱ་བཟང་གིས་བཅད”写成“རྒྱ་བཟང་ཐྱིས་བཅད”。

定义 4 词语错误是指几个正确的藏文音节搭配成词语时，该词语不在藏文词典集合中的无效藏文词语。例如“ང་ན་ཚ་ཨེད། གཡག་ཇ་དཀར་ཤི་སོང་།”写成“ང་ན་ཚ་ཨེད། གཡག་ཇ་ཀ་ཤི་སོང་།”等。一般出现在同音字代替正确字的场合，会导致意思的错误。

定义 5 语法语义错误是指不符合藏文语法结构规律或客观事理的句子错误，包含语法错误和逻辑错误。例如“ཁུ་རྨོལ་ཤིག”写成“ཁུ་བརྨོལ་ཤིག”时态错误等。

根据上述的错误类型，本文从藏文音节拼写检查、梵音转写藏文错误检查、接续关系检查和藏文词语错误检查 4 部分进行探讨，设计相应的藏文文本自动校对系统。

1.2 系统框架

藏文文本自动校对系统框架包含音节的拼写检查、梵音转写藏文检查、接续关系检查、藏文词语校对、语法语义检查等内容。由于词语的错误、梵音转写藏文的错误和接续关系的错误会导致语法语义错误，所以语法语义错误处于系统框架的底部，并与词语错误、梵音转写藏文错误、接续关系错误进行关联。藏文音节作为组成词语单元，它的错误会导致词语的错误，因此音节拼写错误放在词语错误之上，表示音节错误与词语错误的关联，具体系统框架如图 1 所示，其中虚线内是本文讨论的内容。

1.3 系统框架设计

在藏文文本校对系统设计过程中，每个模块有明确的实现功能，但每个模块之间也存在相互依存关系和执行的前后顺序问题。如何确定每个模块之间的顺序是系统设计的关键之一。

藏文文本中一般会出现传统文法规则形成的藏文音节和梵音转写藏文。从表现形式上看，两种字符串都是由音节点来隔开。 在拼写检查时，不能用同一种规则方法来检查拼写的正确性。针对这个问题，首先需要明确藏文音节字符集合和梵音转写藏文字符集合的关系。由于传统藏文文法和梵音转写藏文具有不同的文法体系，因此是两个不同的字符组合关系。但毕竟这两个集合是共同字符的两种不同组合形式，所以两个字符集合是一个大的集合中的不同子集，它们之间存在交集的部分，两个集合关系如图 2 所示，其中 A 是正确的藏文音节集合，B 是正确的梵音转写藏文集合；A 和 B 的交集是共同拥有的正确的部分，图中用交叉斜线部分表示；$A \cup B$ 的补集是藏文文本中字符组合错误的部分。

藏文音节拼写检查和梵音转写藏文错误检查中，需要判断音节点隔开的藏文字符串是否属于 A 或 B 集合。由于一般藏文文本中，藏文音节出现的频率很高，而梵音转写藏文出现的频率很低，如果先检查梵音转写藏文，部分藏文音节作为梵音转写

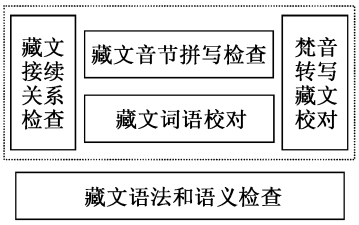

图 1 藏文自动校对系统框架

Fig. 1 Framework of Tibetan automatic proofreading system

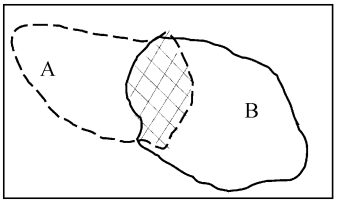

图 2 藏文音节集合与梵音转写藏文集合

Fig. 2 Sets of Tibetan syllable and Sanskrit transliteration in Tibetan

藏文而在接续关系检查中无法检查接续关系。因此, 在检查的顺序上, 本文认为先检查藏文音节, 后检查梵音转写藏文。

从总体框架上来, 藏文文本校对系统通过采用模块化的思想来逐一解决 4 种不同错误类型, 具体实现算法如下, 对应的藏文自动校对流程见图 3。自动校对算法和流程图表示每个模块之间的先后顺序和相互依存关系。

算法 1 藏文自动校对算法。

输入: 藏文文本内容;

输出: 校对结果文本;

1 藏文音节拼写检查, 若拼写正确, 转到 3, 否则, 转到 2;

2 梵音转写藏文错误检查, 若正确, 转到 5, 否则做标记错误, 并转到 5;

3 藏文的接续关系检查, 若接续关系正确转到 4, 否则做标记错误, 并转到 5;

4 藏文分词, 匹配词典, 若匹配成功转到 5, 否则标记错误标记, 并转到 5;

5 输出校对结果。

1.4 系统实现方式

在系统的具体实现过程中, 设计了 8 个类来实现不同的功能。Cheker 类为系统的主类, 衔接藏文音节拼写检查、梵音转写藏文检查、接续关系检查和词语检查。主要功能由 3 个类来完成: SpellCheker 类负责拼写检查, Devanagant 类负责梵音转写藏文检查, SegmentAndWordCheker 类负责接续关系和词语检查。3 个类之间的关系为: 首先, 读取一个藏文文本文件, 并从该文件中按照顺序获取一个藏文音节。其次, 一个藏文音节作为输入条件, SpellCheker 类对该音节进行拼写检查, 如果拼写检查错误, 该音节交给 Devanagant 类来检查梵音转写藏文的正确性; 如果拼写检查正确, 该音节交给 SegmentAndWordCheker 类。然后, Cheker 类中需要累积不低于 4 个的连续音节, 这些音节作为 SegmentAndWordCheker 类处理的对象, 检查接续关系和词语的正确性。在累积 4 个音节的过程中, 出现拼写错误或梵音转写藏文, 处理的对象就按低于 4 个音节来处理。

在藏文音节拼写检查中, 主要有 SpellCheker 类和 Compare 类来完成。SpellCheker 类完成拼写检查的内容, Compare 类实现藏文音节规则模型算法的功能。Devanagant 类完成梵音转写藏文的词典匹配功能, 如果匹配成功, 输出梵音转写藏文, 否则, 输出标记错误的藏文字符串。在接续关系检查和词语检查中, 藏文字符串和位置索引标记 index 作为输入, SegmentAndWordCheker 类完成虚词兼类过滤、匹配格助词和不自由虚词、匹配词语的功能, 负责完成藏文接续关系和词语正确性检查。虚词兼类由 SyllepsesCheker 类来完成, 排除存在歧义的可能性; 接续关系的检查由 JointCheker 类来完成, 按照接续关系检查算法, 检查格助词和不自由虚词接续关系的正确性; 词语检查由 WordCheker 类来完成, 采用

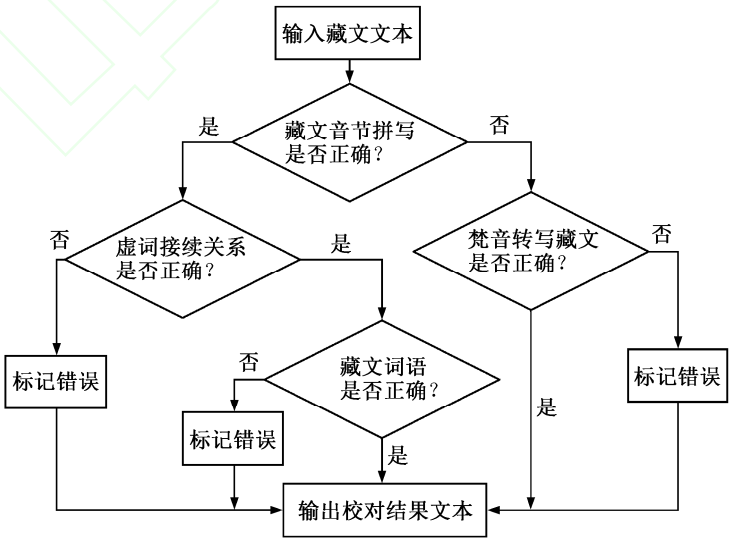

图 3 藏文文本自动校对算法流程

Fig. 3 Algorithmic process of Tibetan text automatic proofreading

正向最大匹配分词算法，检查词语的正确性，与分词不同的是不再进行细切分，而且词条的检查只针对于双音节以上的词汇。系统实现过程的 UML 如图 4 所示。

2 藏文文本校对方法

本节根据各个模块自身的特性，具体探讨每个细节过程，讨论采用的具体方法，并着重讨论接续关系的检查算法。

2.1 藏文音节拼写检查算法

藏文音节拼写检查一般采取两种方法：第一种是收集所有可能的藏文音节，然后采取字典匹配方式进行检查；第二种是采用规则方法来进行拼写检查。本文利用藏文音节规则模型进行拼写检查[4]。

2.2 梵音转写藏文拼写检查方法

梵音转写藏文拼写检查方法中，根据专家整理的 13765 个梵音转写藏文字典为依据，通过采用词典匹配方法进行检查。

2.3 藏文接续关系检查算法

藏文具有丰富的格助词和虚词，其中虚词又分自由虚词和不自由虚词。藏文接续关系中大部分格助词和不自由虚词具有严格的接续规则，不能随意使用接续关系来进行词与词之间的连接。因此，接续关系检查是藏文自动校对中是必不可少的环节，也是与其他语种不同的特有现象。

传统藏文文法中格助词和不自由虚词两种接续关系，其中 5 个属格助词、5 个作格助词、7 个位格助词具有严格的接续关系；在不自由虚词中 3 个饰集词、3 个待述词、11 个离合词、11 个终结词、“ཞིང་”等 14 个虚词、4 个时态助词也具有严格的接续关系。接续关系的定义如下。

定义 6 后缀是指藏文音节的后加字、再后加字、无后加字 3 种类型的字符。

定义 7 接续关系是指针对藏文音节不同的后缀，格助词和不自由虚词严格遵守藏文后接添加规则，该规则称为接续关系。

接续关系是传统藏文文法的组成部分，也是 1300 多年来一致沿用、藏文书写必须遵守的规则。如果不按接续关系来书写藏文，均视为接续错误。接续关系如表 1 所示，该表是根据藏文文法收集和整理的藏文接续关系表。

藏文接续关系用<P, X, f>三元关系模型来进行形式化表示。P 为后缀集合，$P=\{p_i|p_i\in\{$“ག”, “ང”, “ད”, “ན”, “བ”, “མ”, “འ”, “ར”, “ལ”, “ས”, “Φ”, “ ཀྱ”, “ཤ”, “ཧ”$\}$, $i=1, \ldots, n\}$; X 为包含格助词和不自由虚词的集合，$X=\{x_{ij}|x_{ij}\in$格助词和不自由虚词集合，$i=1, \ldots, n, j=1, \ldots, m\}$。$n$ 是后缀字符个数，m 为格助词和不自由虚词个数。f 为接续关系函数，$x_{ij}=f(p_i)$，即某个

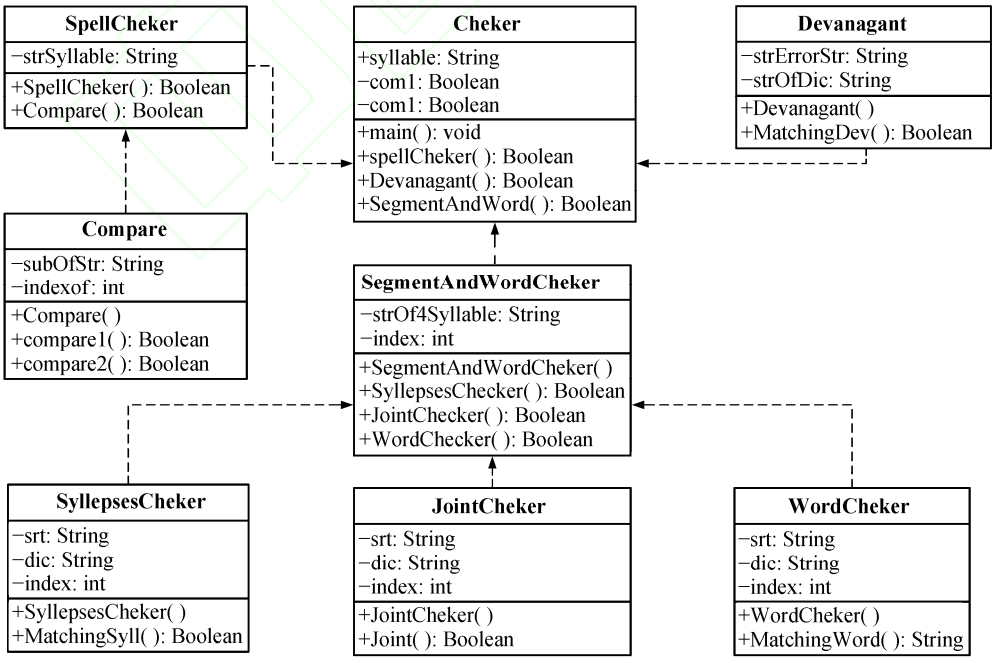

图 4 藏文自动校对系统 UML 图

Fig. 4 UML diagrams of Tibetan automatic proofreading system

表 1　藏文接续关系表

Table 1　Table of Tibetan connective relation

后缀(*P*)	属格助词	作格助词	位格助词	饰集词	待述词	离合词	终结词	时态助词	[ཞེང་]等虚词 (*X*)			
ག	གི་	གིས་	ཀྱི་	གྱང་	ཞེས་	གས་	ཏོ་	གིན་	ཅིང་	ཅེས་	ཅེའོ་	ཅེ་ན་	ཅིག་
ང	གི་	གིས་	ཀྱི་	ཡང་	ཞེས་	ངས་	ངོ་	གིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
ད	ཀྱི་	ཀྱིས་	ཀྱི་	གྱང་	དེ་	དུས་	དོ་	ཀྱིན་	ཅིང་	ཅེས་	ཅེའོ་	ཅེ་ན་	ཅིག་
ན	ཀྱི་	ཀྱིས་	ཀྱི་	ཡང་	ཏེ་	ནས་	ནོ་	ཀྱིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
བ	ཀྱི་	ཀྱིས་	ཀྱི་	གྱང་	ཞེས་	བས་	བོ་	ཀྱིན་	ཅིང་	ཅེས་	ཅེའོ་	ཅེ་ན་	ཅིག་
མ	ཀྱི་	ཀྱིས་	ཀྱི་	ཡང་	ཞེས་	མས་	མོ་	ཀྱིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
འ	ཡི་	ཡིས་	ཡི་	འང་	ཞེས་	འམ་	འོ་	ཡིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
ར	ཡི་	ཡིས་	ཡི་	ཡང་			རོ་	གིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
ལ	ཀྱི་	ཀྱིས་	ཀྱི་	ཡང་	ཏེ་	རམ་	རོ་	ཀྱིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
ས	ཀྱི་	ཀྱིས་	ཀྱི་	ཡང་	ཏེ་	ལམ་	ལོ་	ཀྱིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
ས	ཀྱི་	ཀྱིས་	ཀྱི་	གྱང་	ཏེ་	སམ་	སོ་	ཀྱིན་	ཞིང་	ཞེས་/ ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
无	འི་	ཡིས་	ཡི་	འང་	ཞེས་	འམ་	འོ་	ཡིན་	ཞིང་	ཞེས་	ཞེའོ་	ཞེ་ན་	ཞིག་
	ཡི་	ཡིས་	ཡི་	ཡང་				གིན་					
ནད			ཀྱི་		ཏེ་	ཏམ་	ཏོ་		ཅིང་	ཅེས་	ཅེའོ་	ཅེ་ན་	ཅིག་
རད			ཀྱི་		ཏེ་	ཏམ་	ཏོ་		ཅིང་	ཅེས་	ཅེའོ་	ཅེ་ན་	ཅིག་
ལད			ཀྱི་		ཏེ་	ཏམ་	ཏོ་		ཅིང་	ཅེས་	ཅེའོ་	ཅེ་ན་	ཅིག་

p_i 对应着多个可选的接续关系，只要满足其中一个可选值，就是满足藏文接续关系规则。

藏文音节拼写检查完成后，对正确的藏文音节检查接续关系。从藏文音节结构分析，藏文音节的后缀存在 3 种不同的情况，即 1 个字符后缀、2 个字符后缀和无字符后缀。因此，接续关系检查中首先需要识别集合 *P* 中后缀的不同类型和具体后缀字符；其次需要识别集合 *X* 中格助词和不自由虚词；最后判断是否满足接续关系函数 x_{ij}=*f*(p_i)。根据上面的考虑，藏文接续关系检查算法如下。

算法 2 藏文接续关系检查算法。

输入：输入 srt 和 index

输出：True or False

创建字符串对象 str　//记录读取字符串，字符串的音节长度小于 4

创建整数对象 index　//记录文件读取的索引位置

创建字符串对象 substr　//记录 str 字符串的一个子串

创建字符串对象 x_{ij}　//存储虚词

创建字符串对象 p_i　//存储后缀字符

加载虚词兼类词典和接续关系表

if(substr 是否为虚词兼类)　//过滤虚词兼类情况，虚词兼类匹配成功，返回 true；否则，执行下一步

{index←substr 之后的索引位置

return true

}

else if(substr 是否属于 *X*)　//是否与虚词匹配，如果匹配成功，判断接续关系；否则，执行下一步

{ p_i←从 srt 中取出 substr 之前的最后第 2 个字符

x_{ij}←substr

swich(p_i){　//匹配 1 个后缀字符的情况

case ག: {

if(x_{ij}==*f*(ག)){return true; *index*←*subsrt* 之后的索引位置}

else{return false; index←subsrt 之后的索引位置}

}

……　//按照 ག་ང་ད་ན་བ་མ་འ་ར་ལ་ས་ད་逐一顺序进行比对

case ད:{　//后缀字符为 ད

p_i ←从 *srt* 中取出 substr 之前的最后第 3、2 个字符

swich(p_i){　//匹配 2 个后缀字符的情况

case ནད:{

if(x_{ij}==*f*(ནད)){return true; index←subsrt 之后的索引位置}

else{return false; index←subsrt 之后的索引位置}

}

…

default

return false

}

}

default　//无后缀字符的情况

{if(x_{ij}==*f*(p_i)){return true; index←subsrt 之后的索引位置}

else{return false; index←subsrt 之后的索引位置}

}

}

}

else

{return false; index←subsrt 之后的索引位置}

藏文接续关系检查算法中，为了算法描述的简

便，只检查输入一个字符串的情况，没有加入循环嵌套的过程，但在实现算法时需要考虑循环过程。算法的流程如图 5 所示。

2.4 藏文词语错误检查方法

藏文词语正确性的检查中，通过采用正向最大匹配算法进行词典匹配，检查双音节以上词语的正确性。与分词不同，当词典中的词语不匹配时，不匹配的字符串项不再进行细分，只做错误标记。另外，在词典内容中去除单音节，保留双音节以上的词条。采用 197 个虚词词典、2311 个虚词兼类词典和 133227 个藏文词典。

2.5 测试

测试内容包含接续关系检查算法和系统部分的测试，主要检查正确率、召回率和误判率，并分析每个过程的测试结果。下面对几个测试标准在文本中使用的方法做简要介绍，然后分析实验的测试结果。

2.5.1 评测标准

在文本自动校对中，一般采用召回率、正确率和误判率来评测文本校对算法的性能，具体公式[5]如下：

$$r=\frac{\text{find}}{\text{error}+0.01}, \tag{1}$$

其中 r 为召回率; find 为预校对文本中查出来的真正错误个数，error 为预校对文本中实际错误的个数; 0.01 为平滑系数。

$$a=\frac{\text{find}}{\text{find}+\text{accurate}+0.01}, \tag{2}$$

其中 a 为查准率; accurate 为预校正文本中查出的正确词判错的个数。

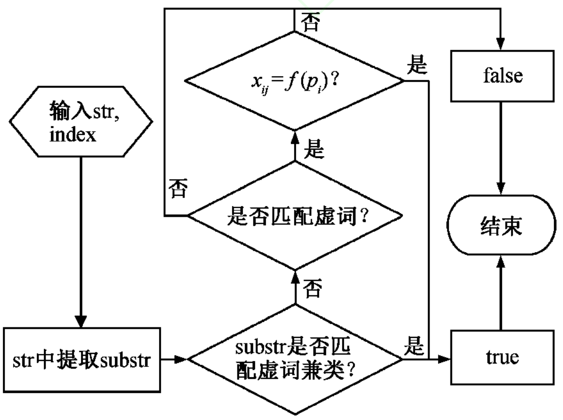

图 5 接续关系检查算法流程

Fig. 5 Algorithmic process of Tibetan connective relation

$$\text{ac}=\frac{\text{accurate}}{\text{find}+\text{accurate}+0.01}, \tag{3}$$

其中 ac 为误判率。

2.5.2 接续关系算法测试

为了测试接续关系算法，本文从“青海藏语广播网”的留言板中收集语料，检查接续关系算法的正确性、稳定性和鲁棒性。由于网上论坛、贴吧、博客等的内容没有经过认真的审核和校对，撰写人员的水平参差不齐，导致文本中经常出现藏文音节拼写错误、接续关系错误、词语错误和语法错误，特别容易产生接续关系的错误。因此，这类语料的测试具有一定的代表性。语料按照不同留言数量，分为步长为 10 的 6 个文件，即第 1 个文件 10 个留言，第 2 个文件 20 个留言等。虽然采用步长 10 来平衡语料，但留言的内容有多有少，很难得到均衡增长的目的。召回率、查准率和误判率测试结果如表 2 和图 6 所示。

从实验中可以发现，接续关系算法的问题主要有以下几种情形，下面通过具体的例子来进行说明。

例 1 紧缩词的识别问题。

格助词和不自由虚词中“འི་འང་འམ་འོ་ས་ར་”紧缩词识别和还原，不仅存在识别的难度，还存在还原的难度，更存在接续关系判断的难度，也是算法召回率

表 2 接续关系算法测试数据

Table 2 Algorithmic test data of the connective relation

文件号	find	error	accurate	召回率	查准率	误判率
1	4	5	6	0.798403	0.3996	0.599401
2	8	10	4	0.799201	0.666112	0.333056
3	19	20	11	0.949525	0.633122	0.366544
4	33	36	23	0.916412	0.589181	0.410641
5	26	30	21	0.866378	0.553074	0.446713
6	23	26	19	0.884275	0.547489	0.452273

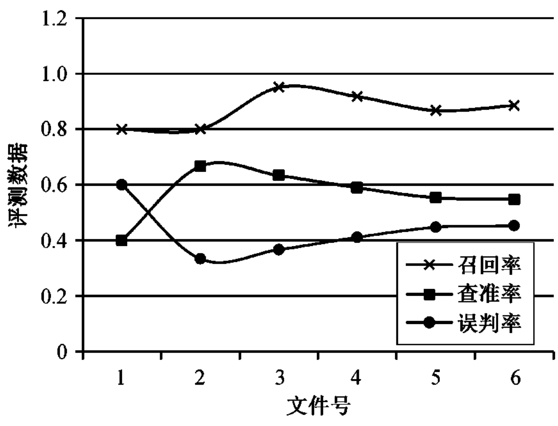

图 6 召回率、查准率和误判率测试结果

Fig. 6 Test results of recall, precision and error rate

和查准率降低的主要原因。例如“ངའི་ཨེ་དཀྱུ་”等。为了解决此问题, 本文将紧缩词的接续关系检查纳入到拼写检查模块中, 然后进行接续关系检查, 但仍然存在“ཚ་ར་”的识别问题。表 3 的数据是改进后的测试结果。

例 2 无后加字的识别问题。

由于音节中没有后加字而算法直接去寻找基字或元音, 如果音节中存在元音或者是纵向叠加情况, 在后加字的判断上就不会存在问题。如果既无元音, 又无叠加情况, 基字又兼后加字时, 算法会在无后加字的判断上存在歧义。例如“ང་ཡིས་ཐུ་གཏང་།”中“ང་”后加字还是基字会出现判断失误。

例 3 两个后缀字符的识别问题。

在两个后缀字符的识别上, 例如“བསྒོད་ཀྱང་”、“ནད་ཀྱང་”中, “ནད་ཀྱང་”按两个后缀字符来对待处理时, 算法对此类语言现象的处理也是存在歧义的。

2.5.3 系统的性能测试

系统性能的测试中, 涉及藏文音节拼写检查、梵音转写藏文校对、接续关系检查、藏文分词等多种校对技术, 每个模块有自身的不足和缺陷, 前面模块的校对结果直接影响下一模块的检查结果, 因此系统的性能受各个模块校对结果的影响, 也受各个模块之间相互关联的影响。对 6 个文件召回率、查准率和误判率语料测试结果如表 3 和图 7 所示。

表 3 藏文自动校对系统测试数据

Table 3 Test data of Tibetan automatic proofreading system

文件号	find	error	accurate	召回率	查准率	误判率
1	14	19	12	0.736454	0.538255	0.461361
2	16	21	11	0.761542	0.592373	0.407257
3	35	41	19	0.85345	0.648028	0.351787
4	60	65	29	0.922935	0.674082	0.325806
5	56	63	34	0.888748	0.622153	0.377736
6	63	70	46	0.899871	0.577929	0.42198

系统测试中存在各类错误, 包括分词错误、接续关系中后缀字符识别错误, 拼写检查错误、各模块交叉错误等。

3 结论与展望

藏文文本校对作为藏语自然语言处理的重要研究内容, 涉及字组织法、词法分析、句法分析等语言学中的主要理论, 不仅有助于藏文自然语言处理理论的提升, 而且在藏文文本检查上有广泛的应用领域。藏文音节拼写检查和梵音转写藏文主要应用藏文字性组织法理论; 藏文词语检查应用藏文词法分析理论; 接续关系检查和语法检查应用句法分析理论和语义学的内容。因此, 藏文文本校对技术的研究能够比较完美地结合 3 个不同层面的理论。另外, 藏文自动校对可以应用在搜索引擎、文字处理、网上资源质量检查等多种领域, 可以提高用户的文字处理效率和文字质量, 提高网上资源的文本质量。因此, 本文以藏文音节拼写检查、梵音转写藏文检查、藏文接续关系检查、词语正确性检查为研究对象, 重点研究了藏文接续关系检查算法、藏文文本自动校对的系统设计, 提出了接续关系检查算法、自动校对的实现框架和算法, 通过实验验证了算法和实现框架的可行性和有效性。本文的研究, 从不同的视角为藏文自动校对提供了实现方法, 从宏观上来说还是属于规则方法的文本校对方法。下一步的工作将继续研究基于统计方法的藏文文本校对方法、基于规则和统计方法相结合的文本校对方法和藏文纠错方法, 为藏文文本的查错纠错提供自动化的处理技术。

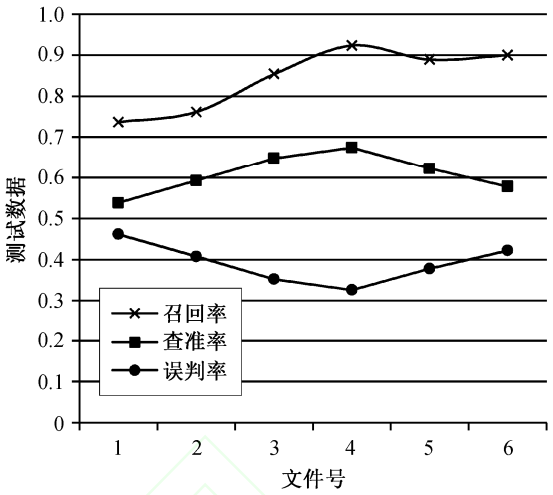

图 7 召回率、查准率和误判率系统测试结果

Fig. 7 System test results of recall, precision and error rate

参考文献

[1] 多杰卓玛. N 元模型在藏文文本局部查错中的应用研究. 计算机工程与科学, 2009, 31(4): 117–119

[2] 刘文香. 藏文文本词校对模型研究. 西藏大学学报: 自然科学版, 2009, 24(2): 70–74

[3] 才让卓玛, 才智杰. 藏文文本自动校对系统开发研究. 西北民族大学学报: 自然科学版, 2009, 30(1): 25–28

[4] 珠杰, 李天瑞, 刘胜久. TSRM 的藏文拼写检查算法. 中文信息学报, 已接受

[5] 张磊, 周明, 黄昌宁, 等. 中文文本自动校对. 语言文字应用, 2001, 1(2): 19–26