一种改进的上下文相关的歧义字段切分算法

A new Context – Sensitive ambiguous phrase segmentation Algorithm

张培颖 李村合 （中国石油大学(华东) 计算机与通信工程学院 东营 257061）

摘要：无论在自然语言处理还是在机器翻译中，中文自动分词都是一个重要的环节。歧义字段切分是中文自动分词研究中的一个“拦路虎”。在分析基于规则和基于上下文的歧义字段切分策略基础上，提出了一种改进的上下文相关歧义字段切分算法，并根据汉语中特殊的语法现象，给出了切分算法的辅助策略来对待切分字符串进行预处理，不仅提高了分词的精度，还加快了分词的速度。

关键词：自动分词 歧义字段 交集型歧义 组合型歧义

1 引言

词是最小的能够独立活动的有意义的语言成分。但中文是以字为基本的书写单位，词语之间没有类似英文空格之类的明显的区分标记。随着中文信息处理工作的推进，对中文自动分词的要求越来越高。无论在自然语言处理还是在机器翻译中，中文自动分词都是一个重要的环节。

歧义字段切分是中文自动分词研究中的一个“拦路虎”。歧义字段分为两种基本类型：交集型歧义和组合型歧义。交集型歧义字段，占歧义字段总数的85%～90%。本文在分析基于规则和基于上下文的歧义字段切分策略基础上，提出了一种改进的上下文相关歧义字段切分算法，并根据汉语中特殊的语法现象，给出了切分算法的辅助策略来对待切分字符串进行预处理，不仅提高了分词的精度，还加快了分词的速度。

2 歧义分析

2.1 歧义产生的根源

分词过程中歧义产生的根源可以归结为以下3个方面：

（1）由自然语言中的二义性所引起的歧义，称为第一类歧义；例如：“美国会采取措施制裁伊拉克”，既可以切分为：“美国/会/采取/措施/制裁/伊拉克/”，也可以切分为：“美国会/采取/措施/制裁/伊拉克/”，两种切分方式在语法、语义上都是正确的，只有依靠上下文和具体的语言交际环境才有可能给出正确的切分；

（2）由计算机自动分词所引起的歧义，称为第二类歧义；例如：“独立自主和平等互利原则”，用正向最大匹配法，就切分为：“独立/自主/和平/等/互利/原则/”，用逆向最大匹配法，则切分为：“独立/自主/和/平等/互利/原则/”，显然在这里，只有第二种切分是正确的；

（3）由分词词典的大小所引起的歧义，称为第三类歧义；例如：“王小二是一个农民”，用计算机自动分词切分为：“王/小/二/是/一个/农民/”，这里“王小二”是一个人名，在汉语中应该是一个词，所以这种切分是错误的。“发展社会主义的新乡村”，“新乡”是一个地名，如果词典中有该词的入口，则“新乡村”就是一个歧义字段。因此，无论词典的大与小都可以产生歧义。

2.2 歧义字段的分类

歧义字段分为两种基本类型：交集型歧义和组合型歧义。下面分别介绍两种歧义字段的定义。定义1（交集型歧义字段）在字段AJB中，AJ∈W并且JB∈W，则称AJB为交集型歧义字段。其中A、J、B为字串，W为词表。

定义2（链长）交集型歧义字段中含有交集字段的个数，称为链长。

（1）当交集型歧义字段具有AJB形式时，只有一个交集字段，称AJB链长为1。

（2）当交集型歧义字段具有ABCD形式时，共有ABC、BCD两个交集字段，称ABCD链长为2。

(3) 在交集型歧义字段中绝大多数为链长1和链长2的歧义字段,二者合计占到了歧义字段的95%。

定义3 (多义型歧义字段)在字段AB中,ABw,Aw,Bw,w为词表,则称AB为多义型歧义字段。如:

(1) A一起领导干部违纪事件。B少年儿童一起拉小提琴。

A、B中"一起"为多义型歧义字段,但A的"一"和"起"都是词,"一起"应该切分,B的"一起"是词,中间则不应该切分。

(2) A人们朝向不同的出口。B他准备的时间不同。

A、B中"不同"为多义型歧义字段,A的"不同"之间不应该切分,B中的"不"和"同"都是词,应该切分。

歧义切分字段还可能有其他混合交叉形式出现,在研究切分技术时,也应该引起注意。如:

(3) A我国目前尚无法人登记法规。B一家人世世代代没有人身自由。

"无、法、无法、法人、人"都是词,组成歧义字段"无法人"。"家、人、家人、人世、世、世世代代、代"都是词,组成歧义字段"家人世世代代"。

2.3 歧义字段的采集方法

由于研究歧义字段是为了正确切分真实的中文文本语料,因此,在大规模的真实中文语料中找出已经出现的歧义字段作为处理对象,这样的处理技术才具有生命力。语料库有生语料和熟语料之分,熟语料就是已经进行切分处理过的语料,便于对比分析。但熟语料规模是很小的,覆盖语言现象不够广泛。因此我们采用从生语料,即未切分处理的原始语料中抽取歧义字段的方法。

目前常用的交集型歧义字段的采集方法是双向最大匹配检索法和正向最大匹配检索与逐词扫描相结合的方法(简称逐词扫描的最大匹配法)。通过分析可知,采用前者可以检查出大多数的交集型歧义字段,而后者可以识别全部的交集型歧义字段。因此我们采用逐词扫描的最大匹配法,其基本过程大致如下:①从被处理文本中的起点取出不超过词典最大长度的汉字串作为匹配字段;②在词典中查找该匹配字段;③如果未找到该匹配字段,则去除匹配字段的最后一个汉字,作为新的匹配字段,并转到步骤(2);④如果找到该匹配字段,则切分出一条词,同时与最近切分的词的做比较;⑤如果二者是交集型歧义字段,根据作出交集型歧义字段的标记,并转到(8);⑥如果二者是组合型歧义字段,则直接转(8);⑦如果二者无歧义关系,则作出词组的标记,并转到(8);⑧后移一个字作为下一次分词的起点,再转到步骤(1)。目前多义型歧义字段的采集方法是通过人工收集,在分词词典中加以歧义标记,然后再利用某些知识来解决。笔者认为可以根据最大匹配法的思想,提出一种基于最大词长的多义型歧义字段识别方法,算法描述如下:①假定分词词典中的最大词长是MAX,则取被处理材料当前字符串序列中的前MAX个字作为匹配字段,查找分词词典,若词典中有这样的一个词,则转②;如果词典中找不到这样的一个MAX字词,则匹配失败,匹配字段去掉最后一个汉字,剩下的字符作为新的匹配字段,进行新的匹配,如此进行下去,直到匹配成功为止;②设匹配字段的字符串S长度为Length,则算法描述如下:

```
for ( int i=1; i<=Length; i++ ) {
  if ( isWord(S.substring(0,i) &&IsWord(S.substring(i,Length)) ) { Flag=true; break; }
}
```

其中:isWord ()是判断是否为词的函数,Flag为是否为多义型歧义字段的标志。

③ 如果Flag为true,则多义型歧义字段被切分出来,剩下的字符作为新的匹配字段,进行新的匹配,如此进行下去,直到所有的字符串都匹配完毕为止。

3 算法描述

传统的切分算法中,无论是正向切分还是反向切分,都会产生一些错误切分,同时也会避免一些错误切分。黄河燕等描述的上下文相关的歧义切分处理算法只考虑到了当前词和前趋词的属性条件[1]。所谓上下文环境,从字面上讲,不应该只有上文环境,还应该具有下文环境。笔者认为加入下文环境的上下文相关的歧义切分处理算法应该更有效。

3.1 歧义规则表示

为了对中文文本中出现的歧义切分现象进行有效的表示,建立简洁逻辑产生式规则描述语言描述歧义切分规则,歧义规则的表示形式为:

<PreAttr> ∧ <CurAttr> ∧ <NextAttr> ∧ <Context>([A][B][C])→<Action>(Flag)

其中：<PreAttr>是前趋词的属性条件描述，<CurAttr>是当前词的属性条件描述，<NextAttr>是后继词的属性条件描述，它们的表示形式为：

<SynCate>(C_1 op_1 C_2 op_2 … op_{n-1} C_n)

<SynCate>是相应词的语法主特征标记，如 NP、VP 等，它可以是 X，表示可以是任意的语法主特征标记，C_1 op_1 C_2 op_2 … op_{n-1} C_n 说明对被测试节点的约束条件集，op_i 是逻辑运算符 AND 或 OR，C_i 可以是下面三种类型：

(1) 数字 表示对被测试词的长度限制；

(2) 汉字字串 限制被测试的词为一个特定的汉语字串；

(3) 属性特征标记 说明对该词的词法、句法、语义属性限制。

在前面还可以加入逻辑非运算子¬，表示不能满足的条件。

<Context>是应用这条规则应该满足的上下文条件，其表示形式为：

<Condition>([A][B][C][Flag])

条件中参数用以限定测试条件是对边的部分或全部属性的检查，其中 A、B、C 分别表示前趋词属性集、当前词属性集和后继词属性集的测试范围，如所有义项和仅一个义项。Flag 可以为 ABSAME、BCSAME，分别表示前趋词和当前词应该是串相等的、当前词和后继词应该是串相等的。

<Action>是满足歧义规则所描述的条件时，应该进行的切分处理函数，包括：

CUT(Flag) 确定一个词的切分

RECUT(Flag) 重新切分

MERGE(Flag) 合并两个词为一个词

DEVIDE(Flag) 将一个词拆分为两个词

3.2 歧义切分算法

上下文相关的歧义处理就是利用歧义切分规则对歧义字段进行切分消歧处理。歧义处理包括：歧义判断、歧义标记和应用歧义规则进行推导消岐。首先，根据上面介绍的歧义字段采集方法判断出歧义字段，做出歧义字段标记，然后如果规则中所说明的前趋词、当前词和后继词的属性特征和上下文条件与当前切分字段的上下文环境条件匹配成功，则执行该规则中所描述的切分处理函数进行消歧处理。

3.3 辅助策略

通过上下文相关的歧义字段切分算法可以有效地处理绝大部分的汉语歧义现象，但汉语是开放的词汇，只有通过知识学习的手段丰富系统的歧义规则，这样的系统才能更健壮；另一方面，在汉语句子中，有时因为修辞、强调和一些细微意义的表达，常常会出现一些词的重叠和一些特殊现象，我们另外增加一些分词知识库，用分词知识库中的知识对待处理字符串进行预处理，这样会避免一些错误的切分，同时也会大大提高分词的速度。分词知识可以描述为以下几种：

(1) 构词知识。构词知识用于构造分词词典中没有的词，可以解决叠词的现象。例如："花花绿绿的世界"，按照汉语的构词法"花花绿绿"是一个词，但分词词典中不可能包括所有形如 AABB 的词，故词被错误地切分。构词知识能够构成形如 AA、AABB、AAB(AB 为词)、ABB(AB 为词)、ABAB(AB 为词)、AXAB(AB 为词)、AXA(A 为动词)、前缀词构成的词、后缀词构成的词等等，有了这些构词知识，这类词就可以正确切分；

(2) 规则知识。从歧义字段形成的词与词之间的结构关系和词性关系出发，总结出一些规则来解决它们，这类知识[4]中作了详细的说明；

(3) 专用知识 。用于正确地解决一个字所形成歧义字段的知识称为专用知识，例如："把"字知识的描述如下：式中 W 为分词词典中词的集合，WD 为动词的集合。

$r=把\ \alpha\beta \cap 把\ \alpha \in W \Rightarrow (\beta \in WD \rightarrow r1=把/\alpha/\beta) \cup (\beta \notin WD \cap \alpha\beta \in W \rightarrow r1=把/\alpha\beta) \cup (\beta \notin WD \cap \alpha\beta \notin W \rightarrow r1=把\ \alpha/\beta)$

利用此知识可以把"把头抬起来"正确切分为"把/头/抬/起来"，"把儿子给你"正确切分为"把/儿子/给/你"，"请拉好把手"正确切分为"请/拉/好/把手"等等。

并不是分词知识越多越好，由于知识之间的相互影响和顺序不同，就可以有不同的切分结果。因此分词知识库应该是开发的系统，用户可以根据实际需要来进行调整、修改、添加等操作。

4 总结

本文对中文自动分词中的歧义字段产生的根源进

(下转第 14 页)

（上接第 48 页）

行了初步分析，重点讨论了交集型歧义字段和组合型歧义字段的定义以及采集方法。在分析基于规则和基于上下文的歧义字段切分策略基础上，提出了一种改进的上下文相关歧义字段切分算法，并根据汉语中特殊的语法现象，给出了切分算法的辅助策略。但如何利用歧义字段的上下文环境等信息、如何完善分词知识库，进一步提高切分正确率有待于更深入的研究。

参考文献

1 黄河燕、李渝生，上下文相关汉语自动分词及词法预处理算法[J]，应用科学学报，1999 年 6 月。

2 温锁林，中文文本歧义字段切分技术[J]，语文研究，2001 年第 3 期。

3 肖云、孙茂松、邹嘉彦，利用上下文信息解决汉语自动分词中的组合型歧义[J]，计算机工程与应用，2001:87－89。

4 梁南元，汉语自动分词知识，中文信息学报，1990(2):29。

5 郑彦斌，书面汉语自动分词及歧义分析，河南师范大学学报，1997(4):25。