

一个基于搜索的中文分词模型

吕强* 钱培德

(苏州大学计算机科学与技术学院)

(江苏省计算机信息处理技术重点实验室)

摘要: 中文分词问题需要在方法上进行新的突破。本文分析了中文分词问题的难点, 提出了中文分词主体不同将导致分词的接受信息量的不同, 从而计算机分词应该注重分词规范的研究。提出了人脑分词的模型, 作为计算机分词仿真的终极目标。提出了基于图搜索的中文分词模型, 将现有分词方法统一在一个框架内, 并且可以很好地承载中文分词问题现有的和潜在的领域知识。指出了中文分词问题是一个多目标优化问题。

关键词: 中文分词, 搜索, 优化

A Search-based Model of Chinese Word Segmentation

Lü Qiang Qian Peide

(School of Computer Science and Technology, Soochow Univeristy)

(Jiangsu Provincial Key Lab of Computer Information Processing)

Abstract: Chinese word segmentation(CWS for short) is expecting new approaches to achieve higher quality. Some critical issues of CWS are analyzed in this paper, and it is pointed out that the difference of the executants of CWS leads to the different input information for the segmentation. Therefore the specification of CWS should be focused on by the research community. A CWS model of human brain is raised, which can be the final target of computer CWS. A search-based model of CWS is proposed. Not only can this model unify the existing CWS approaches, but also integrate and take potential domain knowledge. Finally CWS is interpreted as a Multi-objective optimization problem.

Keywords: Chinese word segmentation, search, optimization

*Email: qiang@suda.edu.cn Tel: 0512-67451516 Addr: 江苏省苏州市十梓街1号158信箱邮编215006

1 中文分词问题再认识

1.1 什么是中文分词问题

我们从简单的几个定义开始形式化描述中文分词问题。

定义 1 字符集 $\Sigma = \{c_1, \dots, c_i, \dots, c_n | c_i \text{ 表示相关的汉字字符}\}$, Σ^* 表示在字符集 Σ 上的字符串集合。

作为字符集 Σ , 对中文信息处理来说, 就是GB2312所对应的字符。从中文分词的概念和技术角度来看, 字符集从GB2312扩展到GBK, 或CJK, 或UNICODE, 或ISO10646, 都对分词问题本身的研究没有本质的影响。

定义 2 针对应用 $d(\text{Application domain})$ 的词汇集 (词库):

$$\mathcal{V}_d = \{w_1, \dots, w_i, \dots, w_v | w_i \in \Sigma^*, \text{且应用 } d \text{ 认为 } Seg_d(w_i) = 1\}.$$

一般地, 我们可以省去针对某种应用 d 的限制, 认为任何词库都是针对一种应用的, \mathcal{V}_d 简记为 \mathcal{V} 。于是 \mathcal{V}^* 表示在 \mathcal{V} 的词汇串集合。

定义 3 分词规范是一个函数 $Seg: \Sigma^* \times \mathcal{K} \mapsto \{1, 0\}$, \mathcal{K} 表示一定的语境, 对于 $w \in \Sigma^*$, $k \in \mathcal{K}$, $Seg(w, k) = 1$ 就表示 w 是词, $Seg(w, k) = 0$ 就表示 w 不是词。一般来说, 当把 \mathcal{K} 退化为一个词库 \mathcal{V} 时, $Seg(w, k) = 1$ iff $w \in \mathcal{V}$ otherwise $Seg(w, k) = 0$ 。

这里, \mathcal{V}_d 强调了依赖于某种应用 d ! $Seg_d(\cdot)$ 与 $Seg(\cdot)$ 既关联又不同, 前者可以来自于后者, 也可以完全不受后者约束。这样, 我们强调了语言的社会属性, 所以认为任何先验的规范 $Seg(\cdot)$ 都不完全限制针对某个后验应用 d 所需要认定的词; 但是, 这种认定在大多数情况下, 应该符合先验规范。从这个角度来看, \mathcal{V}_d 是 $Seg(\cdot)$ 的最简单的一种表达 (representation) 和实现 (implementation)。

区分 \mathcal{V}_d 和 $Seg(\cdot)$ 的重要意义在于, 当我们评价不同的分词方法的时候, 特别是分词结果的时候, 一定要注意作为前提条件的 \mathcal{V}_d 或 $Seg(\cdot)$ 是否一致, 即需要保证 \mathcal{V}_d 或 $Seg(\cdot)$ 的知识来自同样的分词规范。

接下来定义几个操作函数:

定义 4 $tail(s) = tail(c_0 c_1 \dots c_k) = c_k$, $head(s) = c_0$, $cat(c_i, c_j) = c_i c_j$,
 $\forall_{i=0}^k, c_i, c_j \in s$

定义 5 $char(s) = \{c | c \in \Sigma \wedge c \in s\}$

于是, 中文分词问题可以描述为:

定义 6 对于待切分字符串 $s \in \mathcal{S}$, \tilde{s} 是 $s = c_1c_2 \dots c_n$ 的一种分词切分, 记为 $\tilde{s} = c_1 \dots c_i / c_{i+1} \dots c_j / c_{j+1} \dots / \dots c_n /$ 。

定义 7 把分词知识标记为 k , 中文分词问题就是通过计算机求解这样一个映射:

$$k : s \mapsto \tilde{s} \quad (1)$$

1.2 当前的分词方法

一般来说, 中文分词问题的难点表述为: 歧义切分和未登录词问题。但是, 实际上, 中文分词问题的难点还应该加上分词规范这一重点问题。分词规范、歧义切分和未登录词是中文分词问题的三大难点。而且, 其难度性质不同, 问题本身所处的层次也不同。同时, 它们之间又有一定的关联。其中, 分词规范问题是核心问题, 而这恰恰分词研究忽略的问题。

分词问题的解决必然涉及到分词知识的提取和应用, 从这个角度来说, 分词问题是指: 通过分词知识的计算机化, 计算机主体把一个无间隔的汉字串映射到有间隔的汉字词串的问题。

目前所说的分词方法有三大类, 机械分词、统计分词和规则分词。实际上只是公式(1)中 k 的三种表现形式而已。

1. 所谓机械分词, 把 k 实现为一本机器词典, 这样的话, k 这个映射就不能保证对待切分串 s 的唯一分割, 从而导致所谓的歧义切分问题。这是最早的歧义切分的来源。机械分词就是如下这样的映射:

$$k_v : s \mapsto \Psi, \Psi = \{\tilde{s} | \tilde{s} = w_1 / \dots w_n /, w_i \in \mathcal{V}\} \quad (2)$$

很显然, 对于 $\forall s \in \mathcal{S}$, k_v 不能保证 $|\Psi|=1$ 。

2. 所谓统计分词, 就是利用统计信息, 找出 $\tilde{s}^* = \arg \max_{\tilde{s} \in \Psi} P(\tilde{s})$, $P(\tilde{s})$ 是 \tilde{s} 的概率, 可以有各种统计语言模型SLM来计算 $P(\tilde{s})$ 。一般来说, 统计分词都会用到 \mathcal{V} 。虽然有一种特殊情况, “无词典”分词[1], 实际上也是对训练语料库中的词用一种统计规律表示, 从而形成一本不是真正语义意义上的“词典”, 对分词问题来说, 只是 \mathcal{V} 的表达方式不同而已: 即从枚举型表达转化为概率模型产生型而已。
3. 所谓规则分词, 就是首先定义一组分词规则 \mathbf{G} , 它可以是基于词法、句法、语法等等一系列于语言相关的规则, 本质上是描述一个个 pattern。一般来说, 每个 pattern 都可以转化为一棵语法树或者计算机可以处理的数据结构。对于待切

分句子 s , 用 G 去扫描生成语法树之类的数据结构, 如果生成一个合法的结果就是切分结果。很显然, 规则分词不能穷尽(哪怕是覆盖大量的)所有的语言现象, 因为自然语言是一种社会现象, 而不是严格规则现象。同时, 不一致性问题对于规则分词来说, 尤为严重。所以, 需要更多的更大的上下文窗口来消除规则的不一致性。而这些上下文窗口的描述和判断, 又是可能同分词问题难度不相上下的问题。

统计方法, 目前是自然语言处理的主流方法[2], 因为其结果最好。但是, 统计方法的缺点在于:

1. 把语言知识通过处理共现外表(字形或词形)的方式来提取, 显然这样的方法没有考察到字形或词形背后的语义属性, 因而也是不能完全提取语境的信息, 从而不可能根本解决分词问题。所以, 统计方法不应该是唯一可行的自然语言处理的方法。
2. 统计方法对小概率事件固有的偏见, 对于自然语言问题来说, 是一个致命的缺点。在自然语言中, 只出现一次的表达, 与出现千万次的表达, 应该是同等价值的100%正确! 不应该认为出现千万次的共现比出现百十次的共现更能够影响最后的提取的知识。至于说未出现的表达, 它的一旦出现, 也不应该只判定其是否符合或者已经符合多少现有的已经出现的事件。
3. 统计方法不能很好处理远距离的共现问题。由于受到计算复杂度的限制, 统计方法一般只处理近距离的共现, 例如 n -gram模型, 一般实际处理中, 只实现到4-gram, 也就是说, 处理当前字或词的概率的上下文窗口只有最大4个语言单位。而自然语言中, 往往需要完整的语义群作为考察当前字或词的功能属性的背景, 而不是绝对的几个数目的语言单位。

1.3 分词的接受信息量

分词知识的源泉来自于人类的智慧。从某种程度上讲, 分词系统的好坏本质上取决于计算机系统能够承载和实现多少人类的分词知识。我们不妨来看一看人类是如何分词的。

首先, 必须把人类分词至少要区分成两种情形: 对书面汉语的分词和对非书面汉语的分词。区分这两种情形的目的, 在于强调在这两种分词情形下, 人类所接受的信息量是不同的, 从而所使用到的分词知识的方式也有本质不同。

对于非书面汉语的分词情形, 人类的接受的信息是多模 (multi-modal) 且基于理解的 (understandable), 例如, 语音语调、表情、周围场景等等, 都是重要的可靠的分

词信息来源。例如，人们在街上看到一块饭店的招牌，上书“阿三炒饭店”。这时，我们可以不费吹灰之力就可以切分“阿三/炒饭/店”，而不会去考虑是否某位员工“阿三”在“炒/饭店”的鱿鱼。因为，我们在这个场景中接受的信息是如此之丰富：它在一家店的招牌上，我们看到了是一家店铺，可能还有橱窗，我们还看到了食客模样的人进进出出……。

对于书面汉语的分词情形，人类的接受的信息是单模 (uni-modal) 和基于理解的，这些信息是在一定语境中的字符串信息。同样对于“阿三炒饭店”，正确切分为“阿三/炒/饭店”还是“阿三/炒饭/店”，必须取决于上下文的语境和语义信息。如果上下文的语境中，告诉了你，阿三是个人名，且这段文字出现在招牌上(且慢，如果不能理解了“招牌”的含义呢?!)……。

如果把中文分词的主体从人类改变为计算机，那么，计算机所接受的信息就是单模且基于存储的 (storable)，而不是基于理解的。计算机所接受到的字符信息只是面向存储的，本质上说，只是一种字符数据！这些数据必需经过某种表示器处理 (viewer rendering) 后表征给用户的，目前还没有别的附加信息依附在字符串或单个的字符上面，所以，计算机不能理解所有的这些字符数据背后的概念和含义。事实上，要正确理解这些字符数据背后的概念，可能还依赖于分词后的结果！

总结下来，分词规范问题是设计问题，属于概念层上的问题。歧义切分首先是分词规范的实现问题，属于技术层面上的问题。如何完全消解歧义切分，只在分词问题层面上是不可能完全解决的。而未登录词问题则是介于两者之间、或者说两者兼而有之，甚至从某种程度上讲，也可以是分词问题之外的问题。因为所谓未登录，是相对于分词系统所知的词库而言。如果说该分词系统能够在分词知识的层面上识别出一个“未登录”词，那么它就是“词库”中的词，只是表示形式不同而已。而分词规范问题最终转化为分词知识的可计算问题。

重点研究分词规范的意义在于，分词规范实际上是定义“分词问题”这一个题目的唯一表述！不统一在一个分词规范基础上的分词研究，实际上就好比在做不同的题目，那结果比较的意义也就要打折扣[3]。

1.4 人脑分词的模型

我们不妨来讨论一下，人类的分词知识是如何习得的，虽然这只是过程性的描述，本文认为对计算机如何实现分词知识，以及实现哪些类型的分词知识是很有帮助的。

人类的分词知识实际上是其生活知识在语言范畴内的一种表达或者应用，不仅仅是语言范畴内的智能。从低级到高级，人类的分词方法可以描述为：

1. 实例方法(Instance Approach), 或称查表方法。也就是说, 儿童阶段, 对词的认知是依靠被告知, 从而累积了不少实例。通过死记硬背, 精确匹配, 儿童完成了分词第一阶段认识。假定有 $s \in \mathcal{S}$ 的正确切分的实例句子 \check{s} , 那么, 如果待切分句子 s' 与 s 完全匹配, 那么 s' 的切分就使用 \check{s} 的切分结果。这里的一个关键问题是: 如何判定一个待切分字符串? 一般来讲, 应该是由标点符号断下来的自然切分单位。但是, 具体地, 逗号、句号和感叹号等等, 对句断的作用绝然不同, 所以, 如何取得向 \mathcal{S} 匹配的待切分句子 s' , 对计算机来说, 还是一个比较复杂的问题。

例 1 已知 $\check{s} = \text{“研究/生命/运动/现象”}$, 那么, 当 $s' = \text{“研究生命运动现象”}$ 时, 实例方法给出分词结果 $\check{s}' = \check{s}$ 。

2. 模板方法(Pattern Approach), 或称替换方法。把 \mathcal{S} 进行分类, 对具有同一类属性 seg_i 的 s 都具备如下的特性: 其部分成分(一个或多个) w_j 都具有相同“分词”属性, 这个 w_j 可以被替换成其它没有被 \mathcal{S} 包含的实例成分。如果要判定待切分句子 s' 是否可以使用这种模式, 只进行非替换部分的匹配: 例如, 在 \mathcal{S} 中有个 seg_i 的 s 与 s' 匹配, 除了 w_j 部分对应的 w'_j 。而 w_j 和 w'_j 又具有相同的“分词”属性, 那么, 就用这个 s 的切分结果作为 s' 的切分结果。所谓相同的“分词”属性, 是指一些语法句法属性。例如, 相同的词性等等。需要注意的是, 虽然理论上 \mathcal{S} 中的每一个 s 都可以是一个独立的类别, 但是, 需要大量的 s 实例来巩固这个类别可以被将来模板化套用的准确性, 因为进行替换的必要的条件是替换部分的完全匹配。

例 2 已知 $\check{s}_1 = \text{“研究/生命/运动/现象/”}$, $\check{s}_2 = \text{“研究/生命/运动/规律/”}$ 。“现象”和“规律”被归类为分词属性 $w_n = \text{“名词”}$ 的一个分词类 $seg_i = \{ \text{“研究/生命/运动/”} + w_n \}$ 。于是, 当 $s' = \text{“研究生命运动过程”}$ 时, 由于“过程”的分词属性和 w_n 匹配, 并且其它部分都与 seg_i 匹配, 所以模板方法给出分词结果 $\check{s}' = \text{“研究/生命/运动/过程/”}$ 。

3. 概念方法(Concept Approach), 或称抽象方法。这是在 \mathcal{S} 正确切分实例 $\check{\mathcal{S}}$ 中, 切分成分都被抽象表达成语义功能成分(或者可以低级一些, 语法功能成分、句法功能成分、词法功能成分等等), 从而在这些正确切分实例中抽象出一定的模式。待切分句子就在这个模式库中进行匹配, 从匹配到的模式后面的切分实例来推导待切分的结果。这实际上是将正确切分实例和待切分句子按照同一种规则抽象到模式。如果这两种模式一样, 那么使用正确切分实例的结果来切分待切分句子。

例 3 已知 \ddot{s} = “研究/生命/运动/现象/” 被抽象成:

研究类动词+学科类名词+客观型表述名词

那么, 当 s' = “考察队列性质” 时, 由于概念匹配成功, 概念方法给出分词结果 \ddot{s}' = “考察/队列/性质/”。

实际上, 一个成人成熟的分词模式是以上三种方法混合使用的, 并且, 对于一个分词处理实例, 有可能反复多遍地并行交叉应用这些方法。这个理念同危辉的论述一致[4]。这就启发我们, 对计算机的分词来说, 多遍扫描、混合应用多种方法是一个自然选择。而这同进化搜索方法的理念是多么的一致!

经过上述三个方法的反复多遍使用, 人脑的分词知识必然有一个学习过程。也就是说, 如果分词分错了, 人脑应该知道(不一定是实时的, 可以延后一段时间), 并且调整反馈到人脑的分词知识库中。这是一个不可或缺的部分。这个特性要求, 计算机自动分词也必须要有自适应的学习过程。最终表现出来的外在功能是: 即使是对同一个句子, 明天的切分结果可能就同今天的结果不一样, 因为明天将比今天具有更加丰富的经历, 从而学习到了更加准确的分词知识。

综上, 我们总结出最理想的人脑分词模型由下面三条假设构成:

H1 分词方法假设: 实例方法、模板方法、概念方法。

H2 多遍扫描假设。

H3 正反馈学习假设。

图1表示了最高级的分词模型, 人脑输入是待切分的 s , 输出是已经切分的 \ddot{s} 。虚线体现了正反馈学习假设, 左边弧线体现了多遍扫描假设, 在三种分词方法中反复多遍扫描。应该说, 这就是计算机分词需要达到的终极目标。

从计算机科学技术的实现角度来说, 实例方法可以转换为查表处理问题, 模板方法可以转换为统计学习问题, 概念方法可以转换为语义抽取问题, 而多遍扫描和正反馈学习问题可以是一个标准的元启发设计和实现问题。

2 中文分词的搜索模型

2.1 基于图的分词问题模型

把分词问题转化为图的检索问题, 已经有许多成功尝试[5]。这里面向ACO[6, 7, 8]的应用, 略做变形, 成为AntSeg模型[9]。对一个汉字串 $c_1c_2 \dots c_i \dots c_n$, 把每一个汉

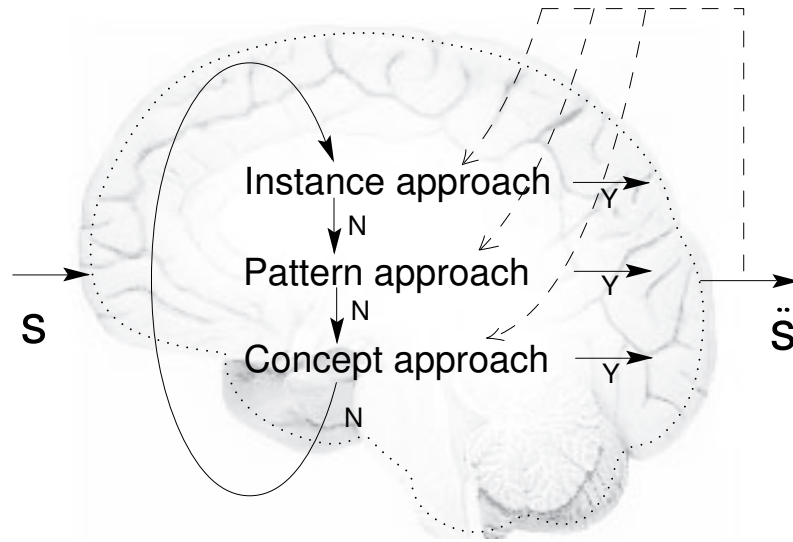


图 1: 人脑分词模型

字都抽象为一个节点。于是首先构成了一个有序图。初始化消歧计算的时候，每一个点上分配一个或若干个搜索智能体。对某一特定的智能体，其 $Open$ 表依次由 c_i 决定。在其搜索方向上，按照一定的规则截取词，然后从所切分的词的下一个汉字开始，重复上述过程，直到把所有汉字节点全部扫描完，称为一代进化。

我们可以把切分串 $s = c_1c_2 \dots c_i \dots c_n$ 转换为面向搜索的图：用头结点 $header$ 指向汉字串链的第一个汉字，还包括汉字串长度。汉字串链将各个汉字节点依次链接为一链表，对每一个节点 $CHChar$ ，如下结构描述：

```
struct CHChar
{
    unsigned char chHead[2];    //对应字符串中的汉字内码
    int icount;                //该汉字为首字的所有可选词的个数
    struct CHWord *WordNext;    //指向词链首
    struct CHChar *ChNext;     //指向切分串的下一个汉字节点
}
```

词链结点 $CHWord$ 结构如下：

```
struct CHWord
{
    unsigned char chWord[MAXLEN]; //词条本身
    struct CHChar *NextChar;      //该词结束后的下一个汉字节点
}
```

```

double icipin;           //词条对应的词频
float pheromone;        //该词上的信息素
struct CHWord *next;    //指向下一个词节点
}

```

以“发展中国家”为例，该切分串图如图2所示。于是，对于切分串中的第*i*个字

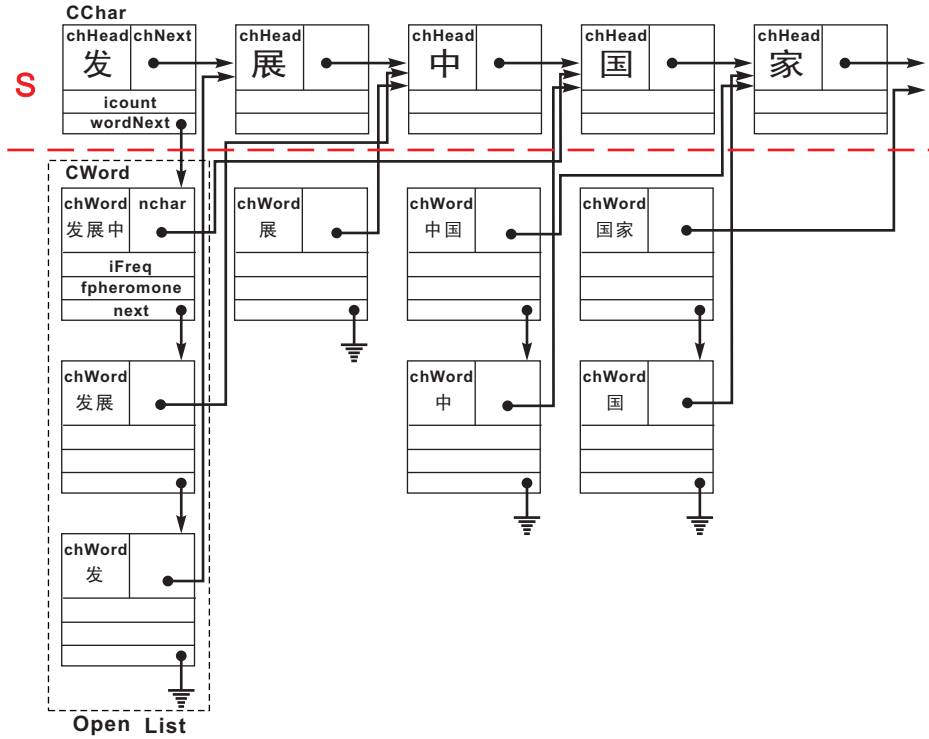


图 2: 面向图搜索的分词描述

符 c_i (CHChar结构的数据变量), 都对应一个 $Open(c_i) = \{w_i | head(w_i) = c_i\}$, 称为字对应的Open表。显然有:

$$icount = |Open(c_i)|,$$

$$c_i.WordNext = Open(c_i),$$

$$c_i.ChNext = c_{i+1}.$$

Open表的构造可以采用如下方法:

- 强单义切分: $\forall c_i \in SegChar \Rightarrow Open(c_i) = \{c_i\}$;
- 根据分词词典, 所有以 c_i 为首字并且在切分串中的词;

- 全切分：以 c_i 为首字到切分串尾所有可能的字的排列，即总共有 $n - i + 1$ 个“词”， n 是切分串的总长度；
- 以上各种都是静态的 $Open$ 表，在开始切分前就可以确定。还可以随着切分进程动态构造。

对某特定的智能体来说，其 $Open$ 表为该智能体所在汉字位置的词链，相应地有 $Solution$ ，依次记录智能体的选词解集。于是，智能体分词的过程是从 c_i 开始，在 $Open(c_i)$ 中选择一个分词结果 w_i ，记载到 $Solution$ 中；再根据 $w_i.NextChar$ 搜索到另一个 c_j ，其中 $tail(w_i) = c_{j-1}, j > i$ 。再在 $Open(c_j)$ 中选择.....，直到把切分串全部处理完毕。这样就将分词问题巧妙地转化为组合优化问题的求解。目标函数值 $J(Solution)$ 将决定哪一个解作为最后的分词解。

2.2 统一的分词模型

我们先把当前的分词方法全部统一到AntSeg解决框架中。

对于机械分词来说， $Open$ 表完全根据分词词典构成。各种方向的扫描完全可以转换为图2中各种链表的构造。智能体的搜索行为很容易实现各种机械分词算法。

对于统计分词来说，各种算法实际上体现为AntSeg目标函数 $J(Solution)$ 的定义，语言模型的选择和训练实际上是离线的。文献[9]描述了如何将n-gram语言模型实现在AntSeg框架下。事实上，AntSeg的迭代特性方便地支持在线的统计学习分词方法。

对于规则分词来说，生成语法树的过程，根本上就可以转化为搜索问题。所以AntSeg很容易成为规则分词的载体。

下面我们简单叙述AntSeg如何处理中文分词的三个难点问题。

分词规范问题和未登录词问题，实际上可以实现在 $Open$ 表的构造和管理上。消歧问题实际上统一在对可行解的评价上，这种评价可以反复多遍的、动态变化的。

AntSeg的另外一个新特性是，支持中文分词的各个部件的并行实现。例如， $Open$ 表的动态更新，多目标评价的并行优化等等。

2.3 分词问题是多目标优化问题

容易明白，对于一个 n 个字符的切分串，所有可能的切分情况是 2^{n-1} 种，发生在 $Open$ 表用全切分构造方法的时候。同时，我们假定：

长距离上下文依赖假定： 正确的分词切分依赖长距离的上下文，可能不止局限于一个句子；

未登录词任意性假定： 所有字符串的排列都可以是未登录词。

在这样两个假定下，分词问题本质上将是一个标准的多目标优化问题 (Multi-Objective problem, MOP)了，且其搜索空间是 $\mathcal{O}(2^{n-1})$ 。

首先一个问题，对于一个中文分词问题， n 有多大？就是问完整切分一个词所需要的上下文有多大？这是一个很艰难的问题。首先，上下文应该定位在义群的层面，而不是段落、句子、字符等表现的层面。不同的文体，如新闻类和文艺作品类，它们的上下文应该统一不到表现层面。但是，义群的划分，也许是比分词问题更加困难的问题。本文认为，这样的上下文粗浅地以3个自然段来划分：上一个自然段和下一个自然段是当前自然段切分的上下文依据。根据对1GB的新闻语料库的统计，自然段落平均长度为41个字符。于是，可以认为， n 的取值可以在100左右。

同时有研究在1991年就表明，95%左右的切分歧义可以借助句法以下的知识解决，只有5%必须诉诸语义或语用知识[10, 11]。多年来没有人对此有新的数据支持或反驳。那么，这5%的语义或语用知识，显然必须通过强大的上下文或者更精确的关于字词的语义知识来获得。这个证据也是这里把中文分词问题转化为组合优化问题的一个依据。

所谓的MOP，形式定义如下[12]。

定义 8 有 k 个目标函数: $f_i: \mathbf{x} \rightarrow \mathbb{R}, \mathbf{x} \in \mathbb{R}^n$ 。最终优化的目标函数是:

$$\min[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \quad (3)$$

当然有 m 个不等约束条件:

$$\forall i = 1, 2, \dots, m \rightarrow g_i(\mathbf{x}) \geq 0 \quad (4)$$

和 p 个相等约束条件:

$$\forall i = 1, 2, \dots, p \rightarrow h_i(\mathbf{x}) = 0 \quad (5)$$

事实上，往往很难有一个 \mathbf{x}^* 能够满足公式(4,5)下令公式(3)的所有 f_i 都达到最小，所以完全是由一个应用问题的情景来确定公式(3)中min的“最优”解释。一般地，有下面定义。

定义 9 搜索空间 \mathcal{S} 是所有满足约束公式(4,5)的 \mathbf{x} 的集合。

定义 10 *Pareto*最优解 (*Pareto optimal*): 存在一个 $\mathbf{x}^* \in \mathcal{S}$ ，如果不存在另外的 $\mathbf{x} \in \mathcal{S}$ 能够令所有 $i = 1, \dots, k$ 的 $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ 且至少有一个 j 使 $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ 。这样的 \mathbf{x}^* 就成为*Pareto*最优解。

Pareto最优解的通俗解释是，再也找不到能够令公式(3)的函数值同时下降的 \mathbf{x} 了。

显然，对于计算机分词来说，目前分词问题不能精确解决。从分词问题的MOP解读来看，可以有下面解释：

1. 切分盲点的存在：由于现实的分词解决方案并没有，也不可能穷尽所有搜索空间。
2. 即使穷尽了所有的搜索空间，也不能有一个目标函数判定正确的分词答案。
3. 可能是分词所需要的语言知识还不能为计算机所获取。

对于第一个原因，解决搜索问题的元启发算法正好是一个解决方案。对于第二个原因，从前面人脑分词的模型来解释，至少，不是用一个目标函数就可以来衡量出分词正确的解，所以多目标函数应该是一个更自然的解决方案。例如，n-gram模型就可以对应多个目标函数 f_i 。但是最终的最优解决不是这些目标函数的简单组合。多目标优化问题的思路应该是解决中文分词问题的新思路。对于第三个原因，更多的是需要领域工程师的工作。

3 结束语

多目标优化应该是中文分词问题解决方案的新思路，但是，由于多目标优化问题本身还有许多挑战，并且都必须是同应用领域紧密相关，所以，对中文分词问题来说，采用基于搜索的多目标优化解决方案，既是挑战，也是机会。另外，并行计算也应该是分词模型应该考虑的计算部件，重要的不是让并行计算加速，而是它将支持某种不确定性的领域知识的处理。面向搜索的中文分词模型可以很好的集成现有的中文分词方法，同时可以承载人脑分词模型的其他逻辑部件和物理实现的计算部件。

参考文献

- [1] 付国宏, 王晓龙. 汉语词语边界自动划分的模型与算法. 计算机研究与发展, 36(9):1144-1147, 1999.
- [2] 黄昌宁. 统计语言模型能做什么. 语言文字应用, 1:77-84, 2002.
- [3] 许顺, 吕强. 试析中文分词国家规范. 中文信息学报, 2006/03. 已投稿.
- [4] 危辉. 基于知觉加工模式的发展式分词算法. 计算机研究与发展, 38(11):1281-1289, 2001.

- [5] 沈达阳, 孙茂松. 汉语分词系统中的信息集成和最佳路径搜索方法. *中文信息学报*, 11(2):34–47, 1997.
- [6] M. Dorigo. *Optimization, learning and natural algorithms (in italian)*. PhD thesis, DEI, Politecnico di Milano, Italy, 1992.
- [7] M. Dorigo, V. Maniezzo, and A. Colomi. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):1–13, 1996.
- [8] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, Advanced topics in computer science, pages 11–32. McGraw-Hill, 1999.
- [9] Qiang Lv, Xiaohu Luo, Peide Qian, and Hongling Wang. Antseg: a new approach to chinese word segmentation. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IEEE IRI 2006: Heuristic Systems Engineering)*, 2006.
- [10] 何克抗, 徐辉, 孙波. 书面汉语自动分词专家系统设计原理. *中文信息学报*, 5(2):1–14, 1991.
- [11] 徐辉, 何克抗, 孙波. 书面汉语自动分词专家系统的实现. *中文信息学报*, 5(3):38–47, 1991.
- [12] C. C. Coello. An updated survey of ga-based multiobjective optimization techniques. *ACM Computer Survey*, 32(2):109–143, 2000.