全切分图与路径表达式在分词算法中的应用

陈晓苏，邹园斌，张文珂

（1. 清华大学物理系 32 班，北京 100084；2. 清华大学物理系 32 班，北京 100084；3. 清华大学自动化系 33 班，北京 100084）

摘　要： 汉语句子 S 的全切分图记作 Graph(S)，意思是，该图的所有路径之集 Path(Graph(S)) 正好表示了 S 的所有切分方案之集 Seg(S). 我们用一个正则表达式 Path-Expression(S)来表示该图的所有路径之集. 因此有 Path-Expression(S) = Seg(S). 然后我们分别给出了分解 Graph(S)与 Path-Expression(S)为素子图（仍然是全切分图）与素因式（仍然是路径表达式）的做法，最后还给出利用全切分图给它的所有路径编码-译码的算法。所有这些想法与做法不仅其正确性可严格论证，而且已设计有算法，并已在计算机上实现了。上述两种素分解能使路径集呈指数性削减，路径表达式全局性地把握路径集，素子图与素因式又都十分简单，可望能给汉语语句的词切分与词性标注，甚至给短语确认和句法成分认定等工作带来积极的影响。

关键词： 全切分图，路径表达式，素分解，路径编码

ASWS-Graph and Path-Expressions in Word Segmentation Algorithm

Chen Xiaosu, Zou Yuanbin,Zhang Wenke

(1.Department of Physics, Beijing 100084; 2.Department of Physics, Beijing 100084)

Abstract: The symbol Graph(S) denotes the ASWS-Graph of a Chinese sentence, the S, where ASWS stands for the set of all scheme of word segmentation of a Chinese sentence. The set of the paths of this graph, which could be expressed as a regular expression, equals to the set of the segmentations of this sentence. I.e. Path-Expression(S) = Seg(S). Graph(S) can be factorized into several prime sub-ASWS-Graphs and Path-Expression(S) can also be factorized into several prime factors by using ASWS-Graph. In the end, we present a method of encoding and decoding of all paths of the graph, by using ASWS-Graph. All conceptions and methods mentioned above can be not only proved strictly, but also implemented by quick programs.

Keywords: ASWS-Graph; Path-Expression; Prime decomposition and factorization; decoding & encoding of paths

1. 引言:

汉语自然语言处理遇到的第一个难题就是词的切分。切分问题已经研究了二十多年，提出了种种策略，用上了形形色色的资源与技巧，始终不能最后解决问题。正如[1] 所指出的，按词语计，即使分词准确率高达 98%以上，然而，从语句看，其准确率却只达到不足 87%. 更因为，分词的错误将使后续的词性标注、短语确定、句法成分确认、语义表达式生成等一系列努力的成效大打折扣。因此有学者提出所谓“一体化”的应对策略，以便借后续的句法分析、语义表达等工作来协助选取满意的切分方案，比如见[2][5]与[3]。[2](与[5])的做法是，让词切

基金资助：清华大学计算机系邓志东老师 SRT 项目：“句法语义一体化方案研究”；

作者简介：陈晓苏(1984-)，男，南京，本科在读，Email：czs@s1000e.cs.tsinghua.edu.cn

邹园斌(1986-)，男，江西，本科在读，Email：zyb081203@sohu.com

分与词性标注模块一次“输出 N 个认为是最优的切分标注结果。”然后，句法分析系统将其加工成结果：“M 个最优的句法树”（[2]，p604），输出给拟议中的语义分析模块。意图是通过提供多个切分-标注方案以补救正确率不足。这种做法只是部分地一体化。[3]则主张实现“字-词-（语）块-句-（语）段”的一体化分析（[3]，p45）。其设想是，在分词-词性标注阶段就尽量争取用上部分后续工作的局部结论。也就是将分词与词性标注工作完成于整个语言分析过程之中，这是真正的一体化分析的做法，是真正的出路。

全切分的做法最切合这两种一体化策略的需要。全切分算法意在保有全部切分信息，比如见[4]. 其困难是全部切分方案非常巨大，使算法不实用，比如见[5][6][8]。因此如何组织与表示全部切分方案就成了一体化做法能行与否的最具决定意义的关键了。本文提出全切分图及其分解、路径表达式及其分解、路径编码-译码等的一整套快速算法，着眼于如何生成、表示、组织、管理与应用全切分集，其意在支持上述的一体化做法，特别是[3]的主张。

2. 全切分概念、切分图 SDAG 与路径表达式

定义 1　给定词表 Γ. 汉字串 $S=c_1c_2\cdots c_{n-1}c_n$ 按词表 Γ 的一个切分方案是把 S 分隔成若干段，使每段都是 Γ 中的词语。于是得出一个词语序列，就用该序列表示该切分方案。

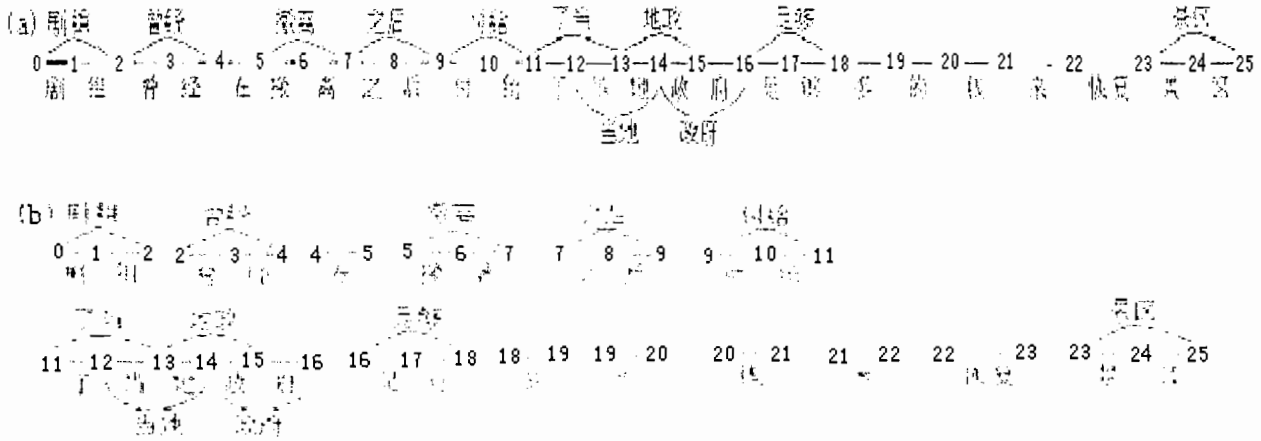
一个语句或任何汉字串 S 的所有的可能的切分方案的集合记作 Seg(S)，叫做 S 的全切分集。Seg(S)的势# Seg(S)就是语句 S 的所有可能的切分个数，叫做 S 的全切分数。

我们主要关心满足 $Seg(S)\neq\phi$ 的汉字串。称这样的汉字串为合适字串。

一个切分方案可理解成其构成词语的序列。对之可形象地理解成从句首到句尾按顺序相继联成首尾相接的几条弧作成的一条路径，每条弧上标以对应的词表 Γ 上一个词语。下面我们将一种切分方案混同于它的路径表示，统称为切分方案，或简称作一种切分。

把汉字串 S 的所有切分方案对应的路径绘在一起得到的图叫做 S 的全切分图，记作 Graph(S). 这是一种有向无圈图(DAG, 比如参看[5].)图 1 是个例子。图 1(a)是个全切分图。

例 1　S：剧组曾经在撤离之后付给了当地政府足够多的钱来恢复景区



(c) (剧*组+剧组)*(曾*经+曾经)*在*(撤*离+撤离)*(之*后+之后)*(付*给+付给)*(了*(当*(地*(政*府+政府)+地政*府)+当地*(政*府+政府))+了当*(地*(政*府+政府)+地政*府))*(足*够+足够)*多*的*钱*来*恢复*(景*区+景区)

(d) $\#Seg(S_2)$：　$2*2*1*2*2*2*(1*(1*(1*2+1*1)+1*2)+1*(1*2+1))*2*1*1*1*1*1*2=2^7*8=2^{10}=1024$

图 1 全切分图　素分解　路径表达式　路径计数

Fig.1 ASWAS graph and prime decomposition of it, path-expression, total number of the paths

2.1 全切分图的生成

全切分图(V,A)是一种特殊的 DAG：它是连通的，恰有一个根（叫作源）$s\in V$，一个汇 $c\in V$，且是可线性化

的，即，它的结点集 V 上可定义一个全序≺，满足条件：∀a∈A. tail(a) ≺head(a)[1]. 这样的 DAG 称为流图，可完整地记作 G=(V,A;s,c,≺).下面约定几个记号。

设 a∈V，记 Out(a)={e∈A |tail(e)=a}, In(a)={e ∈A |head(e)=a}; Succ(a)= {head(e) |e∈ Out(a)}, Pred(a)={tail(e)|e∈ In(a)}；分别叫做 a 的入弧集与出弧集,和后继集与先行集.

定义 2 设 $S=c_1\cdots c_n$ 是 n 个汉字 c_j 的序列，我们用 $S[j, k] = c_jc_{j+1}\cdots c_{k-1}c_k, j \le k$ 表示汉字串 S 中从第 j 个元素 c_j 到第 k 个元素 c_k 的连续 k-j+1 个汉字组成的子串。

记 $P=\{i \mid 1 \le i \le n, Seg(S[1,i]) \neq \phi\}$，叫做 S 的生长点集。记 P+1={i+1| i∈P},

对任意 j∈ P+1，分别引入下面四个集合：

记 $S(j)=\{S[j, k] \mid S[j, k] \in \Gamma, j \le k\}$，叫做 c_j 的 S-生成元集，简称作 S-集，其元素叫 S 弧；

记 $T(j)=\{k \mid S[j, k] \in \Gamma, j \le k\}$，叫做 T-集；

记 $Q(j)=\{S[j, k] \mid S[j, k] \in \Gamma, j \le k, Seg(S[k+1,n]) \neq \phi\}$，叫做 c_j 的 Q-生成元集，简作 Q-集，其元素叫 Q 弧；

记 $R(j)=\{k \mid S[j, k] \in \Gamma, j \le k, Seg(S[k+1,n]) \neq \phi\}$，叫做 R-集;

其中，T(j)表示汉字串 S 中从 c_j 起向右延伸能成为 Γ 中的词语的位置之集。R(j)是其中有路径能延伸到字符串 S 的尾的那些位置。

算法 1.1 全切分图的生成: 从长为 n 的汉字串 S 计算出它的全切分图 Graph(S)

为了简化算法表述，假定生成的切分图的结点集 V={1,2,⋯,n},其全序≺就是普通的<.

1. 生成 T-图 T-G(S);

算法核心是借用 S 集与 T 集，生成一个根为 1 的由 S 弧做成的树，叫 T 图 T-G(S).

2. 修改成 R-图;

按照 Q 集与 R 集修改上述的 T 图，使只保留 Q 弧，删除可能有的所有孤立结点，就得到 R 图 R-G(S), 即 Graph(S).

运用定义 2 引进的概念，很容易证明上述算法的正确性：Seg(S) = label (Path (Graph (S)))[2].

为了克服全切分集巨大带来的困难，我们引入全切分图的素分解与路径表达式的概念。图 1(b)(c)是例句 S 的全切分图的素分解与其对应的路径表达式。(d)是其路径计数。

2.2 全切分图的素分解

全切分图是个流图 G=(V,A;s,c,≺)，我们来考查结点的全序集(V,≺).

定义 3. 设 U⊂V, 记 Out(U)={e∈A |tail(e)∈U, head(e)∉U}, In(U)={e∈A | head(e)∈U, tail(e) ∉U }与 Succ(U)={head(e) ∈V|e∈Out(U)}, Pred(U)={tail(e) ∈V|e∈In(U)}；

定义 4. 设有流图 G.=(V,A;s,c,≺). 设其结点集(V, ≺)可表示成序列：$v_1=s,v_2,\cdots,v_{n-1},v_n=c$;

设(U,≺)是 V 的一个子段（叫区间）：$v_j,\cdots,v_k$; 如果下述性质成立，我们说，在流图 G 中 U 是 V 的一个完美区间：$Pred(U)=Pred(v_j)$, $Out(U)=Out(v_k)$;

完美区间恰有一个入口，一个出口。不含完美子区间的完美区间叫作极小完美区间。

定义 5.（流图的⊙分解）设 G=(V,A;s,c,≺)是一个流图。设 $U=v_j,\cdots,v_k$ 是全序集 $V=v_1,\cdots,v_n$ 的一个完美区间。记 $U^-=v_1,\cdots,v_j$,与 $U^+=v_k,\cdots,v_n$ 是 V 被 U 分割成的另两个区间。定义 $G_0=Gl_{U^-}$, $G_1=Gl_U$, $G_2=Gl_{U^+}$[3]; 称作 U 决定的 G

---

[1] 对任意弧 e∈A，设 e=(a, b)，我们记 tail(e) =a, head(e) =b.

[2] 符号 label(e), label(p), label(T)分别记弧 e 上的词语、路径 p 上的词语串、由路径集 T 的所有路径上的词语串做成的集合。

[3] 符号 Gl_W 表示 W⊂V 在图中确定的子图。注意，U^-与 U，U^+与 U，分别有公共结点 v_j 与 v_k.

的⊙分解。如果①j=1, 或②j=k, 或③k=n，则①G_0、或②G_1、或③G_2分别退化成仅含一个结点的图（简称作单点图，记作 1）

不能再做⊙分解的流图叫做素流图。上面定义中的 U^-与 U^+也是完美区间；三个图都是流图：$G_0=(U^-,Al_U^-;v_1,v_j,\prec)$, $G_1=(U,Al_U;v_j,v_k,\prec)$, $G_2=(U^+,Al_U^+;v_k,v_n,\prec)$; 且有 $G=G_0\odot G_1\odot G_2$;此处二元运算 $G_1\odot G_2$是把 G_1的汇与 G_2的源相衔接;单点图 1 是⊙乘的单位元:$1\odot G=G\odot 1=G$..

算法 1.2(⊙素分解法)：给定素的流图 G=(V,A;s,c,≺)；

算法的核心是逐次寻找(V,≺)最左端的最小完美区间。设 start=1,找出最小 end,使 U=[start,end]满足 Pred(U)=Pred(start),Out(U)=Out(end);U 就是第一个最小完美区间。令 star=end,再找出下一个最小完美区间。如此反复，直到 end=n 为止，否则流图 G 出错。

2. 3 路径表达式的生成

易证，$label(Path(G_1\odot G_2))=label(Path(G_1))\bullet label(Path(G_2))$. 式中•是符号串集的接续运算。故而只需生成素流图的路径表达式。与原切分图相比，素子流图的路径集有指数量级削减。而路径表达式更使所有路径浓缩成一个正则表达式。这样可望全切分算法变得实用起来。

算法 2（借助素切分图计算出路径表达式）给定素的流图 G=(V,A;s,c,≺)；

算法分正向反向两步遍历结点集 V. 引用数组 term 存放中间结果。

1.从结点 s 按序≺直到结点 c,逐个处理 V 的每个结点 a：将 Out(a)存入数组元素 term[a]；

2.从结点 c 按序≻直到结点 s,逐个处理 V 的每个结点 a: 修改 term[a]的内容如下：对每个 e∈Out(a),将 term[head(e)]接在 e 后面；

3.最后结果存在 term[s]里。

2. 4 路径编码与译码

我们构造一个可逆函数 f，给全切分图 G 中的所有路径，从 0 到 n–1 顺序给 n 条路径编码，其中 n=#path(G). f 之逆记作 g.其实，任何互逆的函数 f 与 g 都能作为编码-译码函数，且编码函数 f 的值域，即译码函数 g 的论域，可以是任意的。

我们分三层实现编码-译码函数。设①一个汉语句子被一些特殊字符，如标点符号、数字、外文字符等，分隔成若干子段；②每个子段被剖分成若干个极小完美区间；③每个极小完美区间确定一个素流图。算法的中心是③，借助素流图作出编码-译码函数 f, g; 假定图中每个出弧集都是全序集。对应①、②，我们借用已有的变进制计数法进行编码-译码。

算法 3 路径编码-译码（篇幅所限，从略）

3. 可能的应用概述　代结束语

下面是文方法的可能的若干应用，是我们已经做、正在做或将要做的。

①路径计数与枚举.(1)将路径表达式的各个原子项全都代成数 1，得到一个算术表达式，该式的值就是路径总数。路径计数已经用在我们的路径编码与译码算法里了。(2)在全切分图每个结点的出弧集内排上顺序；然后从源结点 s 开始按深度优先遍历整个全切分图，就得到全部路径（枚举）。按适当的方法给每个出弧集排序，比如给同一出弧集的每个弧赋值，再按权值排序，就可以按优先顺序枚举路径。由此可得到一种最粗的选优算法。考虑到素分解是一种正交分解，可以对各个素子图采用不同的赋值方法，这样，就能使该选优策略更为灵活。(3)而

且，运用路径编码-译码算法，还可以从任意路径或路径序号开始，枚举相继的任意多条路径。这对更新路径，即从已选路径出发进行修正的做法，特别有意义。

②分词算法研究.(1)凭借我们的算法的快速，我们考查了大型语料中交叉歧义、组合歧义的分布情况。从素分解实例中随机抽取样例加以分析可以得到许多有趣的结论。比如，“素分解的正交性和素切分图的信息完全性能帮助我们较好地形成有效的语境。”等等。表 1 给出了素子图按结点数大小的计重分布情况：单字词约占 40%，双字串占 50%多，其中有组合歧义的也容易消解。其它含交叉歧义的多字串，也大都可借语境信息简单地排歧。因此，粗糙地讲，如果采用先易后难的顺序，而不是从左至右或从右至左的固定顺序，再有效利用逐步形成的语境，可望形成有效的分词算法。这里用到一体化的做法。等等。内容较多，也值得进一步分析。(2)同样利用上述的正交性与信息完全性，可望改进 N-最优算法。

③上述正交性和完全性也可能有助于尝试辨识某些语缀、某些句型与句式、以及一些固定结构，像数量结构、介宾结构、趋向结构等等需要处理跨距离的联系结构。我们正在尝试。值得一提的是，各种词组、短语结构、固定搭配等均可以当作特殊弧添加到全分图中。

④我们最关心的是推进一体化分析的策略。在完成上述几项字目标后，终极目标就不远了。

表 1.素切分图的分布情况(计重)

Tab.1 Distribution of the segmentation graphs

结点数	1	2	3	4	5	6	7	8	9	10	11
出现个数，计重	25987	33158	3644	2292	301	85	47	14	2	4	1

鸣谢：感谢审稿人提供重要参考文献和宝贵意见。感谢计算机系陈祖舜副教授给予悉心指导。感谢大会组委会提供资助，SRT 项目提供基金支持。

参考文献：

[1] 曲维光，分词系统计量研究与改进方案，*SWCL2004*(85-90)

[2] 江丰、刘慧、陈玉泉、陆汝站，一个可扩张的汉语词法和句法分析一体化系统，*JSCL-2005*(603-605)

[3] 周强，基于语料库的汉语句法分析和知识获取研究，*中文信息处理若干重要问题*，（34-47）

[4] 孙斌，切分歧义字段的综合性分级处理方法，*北京大学计算语言学研究所讨论班* 1999.4.13,(转引自[5])

[5] 张华平，刘群，基于 N-最短路径方法的中文词语粗分模型，*中文信息学报*，vol. 16. no.5., 2002(1-7)

[6] 蔡勇智，基于最大匹配分词算法的中文词语粗分模型，*福建电脑*, 2005(9)(39-40)

[7] 于源，衣袭，中文全切分快速分词方法，*大连铁道学院学报*,vol.26.no.2.2005.6.(84-85)

[8] 徐华中，徐刚，一种新的汉语自动分词算法的研究和应用，*计算机与数字工程*, vol.34.no.2. 2006 (135- 138)