基于概率和句法分析的中文句子修剪 *

陈劲光 [1),3)] 何婷婷 [2)] 李芳 [1)] 桂卓民 [2)]

[1)] (华中师范大学教育信息技术工程研究中心 武汉 430079)

[2)] (华中师范大学计算机科学系 武汉 430079)

[3)] (湖州师范学院教师教育学院 湖州 313000)

Email: cjg2003@hutc.zj.cn, tthe@mail.ccnu.edu.cn, Wenzheng38@sina.com, fang_lf@163.com

摘要: 提出了一种中文句子修剪方法。引入噪音通道模型，经过改进，提出了更适合句子修剪任务的 Bi-NC 模型。引入无导的方法，解决了中文中缺乏原句-压缩句对齐语料的瓶颈问题。提出了一种自底向上的层级优化算法，避免在优化过程中删除最优修剪句，解决了长句处理时间过长的问题。实验结果表明，本文提出的中文句子修剪方法获得了较好的效果。

关键词: 句子修剪，噪音通道模型，句法分析，中文句子修剪

A Probabilistic and Syntactic Approach to Chinese Sentence Compression

CHEN Jinguang [1),3)] HE Tingting [2)] GUI Zhuoming [2)] LI Fang [2)]

[1)](Engineering & Research Center for Information Technology on Education, Huazhong Normal University, Wuhan, 430079)

[2)](Department of Computer Science and Technology, Huazhong Normal University, Wuhan, 430079)

[3)](School of Teacher Education, Huzhou Teachers College, Huzhou, 13000)

Email: cjg2003@hutc.zj.cn, tthe@mail.ccnu.edu.cn, Wenzheng38@sina.com, fang_lf@163.com

Abstract: In this paper, we describe an efficient probabilistic and syntactic approach to Chinese sentence compression. We introduce the classical noisy channel model into Chinese sentence compression and improve it in many ways. Since there is no parallel training corpus in Chinese, we use the unsupervised method. This paper also presents a novel bottom-up optimizing algorithm which considering both bigram and syntactic probabilities for generating candidate compressed sentences. We evaluate results against manual compressions and a simple baseline. The experiments show the effectiveness of the proposed approach.

Keywords: Sentence compression, Noisy channel model, Syntactic analysis, Chinese sentence compression

1 引言

句子修剪也称为句子压缩或句子约简，是指用相对简略并合乎语法的形式表达一个句子的核心内容。句子修剪可以被用于文本摘要、电子邮件提醒、聚合内容（Really Simple Syndication, RSS）、信息检索、语音识别、移动

* 本文承国家自然科学基金重大研究计划（90920005），国家自然科学基金（60773167），国家十一五科技支撑计划课题（2006BAK11B03），973 国家重点基础研究发展计划（2007CB310804），教育部/国家外国专家局高等学校学科创新引智计划（B07042），湖北省自然科学基金计划项目资助 （2009CDB145）和武汉市晨光计划项目资助（201050231067）的资助。

通讯等多个领域。例如在移动通讯领域，将修剪以后的文本内容呈现在相对很小的手机屏幕上可以让用户浏览到更多的信息。这些领域中普遍存在信息表达简约化的问题，简化版本的句子可以使信息传递更加快捷、迅速。句子修剪研究对自然语言处理领域的发展有重要意义，应该引起必要的重视。在英文领域，句子修剪是自然语言处理的基本问题之一。而在中文领域，虽然已有不少专家开展了句法分析方面的研究，但就我们已阅读的文献来看，还没有发现中文句子修剪方面的论文。

本文第 2 节介绍句子修剪的相关研究工作；第 3 节介绍了中文句子修剪方法；第 4 节介绍了本文方法所取得的效果及分析。

2 相关研究工作

句子修剪的方法主要有两类：基于人工规则的和基于统计机器学习的。

基于人工规则的方法是指按照人工制定规则删除句子中特定的成分，例如字、词、短语、块（Chunk）等。这其中主流的方法是，对原句进行句法分析，生成句法树，根据规则剔除次要的短语或从句。单文档文摘系统 Cut-and-Paste[1]以及多文档文摘系统 SC[2]以及 CLASSY[3][4]中采用了基于人工规则的方法。基于人工规则的方法的关键问题在于人工规则的选取。规则过强会显著降低正确率。通常采取谨慎的做法，增加约束条件，只删除最有把握删除的内容，以提高正确率。基于人工规则的方法存在的根本问题是对于人类知识的依赖，带有很大的主观性。在实际操作中，由于语言表达的多样性，要制定合适的人工规则非常困难。

基于统计机器学习的方法可以在一定程度上弥补基于人工规则的方法的不足，用机器学习来的知识取代人类知识。按照训练语料的类型，基于统计机器学习的方法又可以被分为三类：有导、无导和半指导的方法。有导的方法一般是指使用了“原句-压缩句”对齐语料作为训练语料的方法。Jing[5]提出了一种利用多个知识源来决定句子中的哪个短语应该被修剪的方法。Knight 和 Marcu(以下简称 K&M)[6]提出了采用机器学习进行句子修剪的方法，分别采用基于噪音通道模型的方法和基于决策树的方法解决句子修剪问题，为后来许多研究奠定了基础。各种判别模型被用来降低机器学习的错误率，例如最大熵方法、支持向量]、最大边缘（Large-Margin）方法等。相对于有导的方法，无导的方法是指没有使用对齐语料的方法。而半指导的方法则同时采用了有导和无导的方法。 Hori 和 Furui[7]采用一种无导的方法来删除句子中的词语，利用动态规划算法优化句子修剪过程，但他们的方法没有考虑句法方面的信息。Turner 和 Charniak (以下简称 T&C)[8]改进了 K&M 的方法，并且分别采用有导、无导、半指导（Semi-supervised）三种方法解决句子修剪问题。文献[9]采用整数规划算法对句子修剪过程进行全局优化。

3 中文句子修剪方法

基于 K&M 和 T&C 的思想，本文提出的句子修剪方法的框架由训练过程和修剪过程两部分组成。图 1 给出了中文句子修剪系统的流程图。

训练过程首先下载相关门户网站的语料，经过预处理，构建句子修剪语料库；随后利用开源工具对句子修剪语料库进行分词和句法分析，构建大规模的中文宾州树库；然后由句子修剪语料库导出 bigram 频次表和规则频次表。

修剪过程首先对原句进行中文分词和句法分析，生成句法树；随后利用句子生成器生成多侯选句，形成句法森林；最后通过 Bi-NC 模型，结合自底向上的层级优化算法从句法森林中选出最优修剪句。

3.1 K&M 方法

噪音通道模型是一种概率方法，被成功应用于语音识别、机器翻译、词性标注、信息检索等多个领域。K&M 将这种方法引入到句子修剪问题中，将一个长串看作是 1）以前曾经是一个短串；2）有人人为地在这个短串中加入了不必要的、可删除的成分（噪音）才构成了长串。

句子修剪被看成是从长串中去除噪音、识别原始短串的过程。给定长句子 l，句子修剪任务可以被看成是寻找短句 s，使得 P(s|l)最大。

根据贝叶斯公式：$P(s \mid l) = \dfrac{P(l \mid s)P(s)}{P(l)}$

对于所有可能的短句 s 来说，P(l)都是相同的，因此可以忽略。P(s)是源模型，表示的是短句 s 作为一个句子而存在的概率。P(l|s)是通道模型，表示长句子 l 由短句 s 拓展出来的概率。噪音通道模型将句子修剪问题转变为两个独立的问题：短句 s 是否合乎语法（P(s)），以及短句 s 有多少可能拓展出长句 l（P(l|s)）。结合源模型和通道

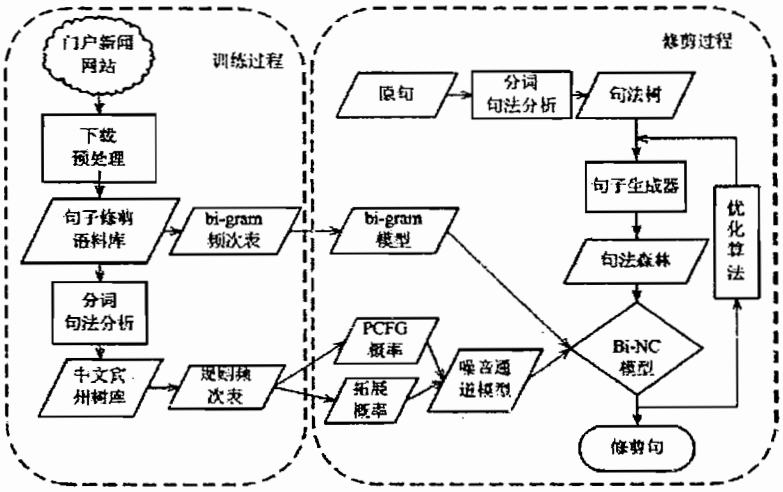

图1 无导中文句子修剪系统

模型找出 P(s|l)的过程被称为解码器。源模型、通道模型、解码器组成了噪音通道模型的三个要素。

在 K&M 方法中， 句子经过句法分析，以句法树的形式呈现出来。源模型由 1）句法树的 PCFG 得分；2）句子中词语的 bigram 得分共同表示。 例如，对于句法树：*(ROOT (IP (NP (PN 我)) (VP (VV 看见) (NP (PN 他))))*

其源模型概率可以表示为：$P(s)=P_{cfg}(ROOT\rightarrow IP|ROOT)\cdot P_{cfg}(IP\rightarrow NP\ VP|IP)\cdot P_{cfg}(NP\rightarrow PN|NP)\cdot P_{cfg}(PN\rightarrow 我|PN)\cdot P_{cfg}(VP\rightarrow VV\ NP|VP)\cdot P_{cfg}(VV\rightarrow 看见|VV)\cdot P_{cfg}(NP\rightarrow PN|NP)\cdot P_{cfg}(PN\rightarrow 他|PN)\cdot P_{bigram}(我|BOS)\cdot P_{bigram}(看见|我)\cdot P_{bigram}(他|看见)\cdot P_{bigram}(EOS|他)$

对于通道模型，为了找出 P(l|s)，K&M 方法通过将对齐语料中的短句和长句中的节点对齐来找出对齐事件，利用对齐事件来计算短规则rs 到长规则r1的拓展概率。即：$P(rl\ |rs\ )=\frac{Count(jo\,int(rl,rs))}{Count(rs)}$

其中 $Count(jo\,int(rl,rs))$ 是对齐事件的频次。K&M 方法使用一个句子生成器[10] 来生成所有可能的候选修剪句，并利用噪音通道模型来进行解码，从而选择修剪句。

K&M 方法要引入到中文句子修剪中，需要解决三个主要问题：1）中文无对齐语料的问题，本文通过引入 T&C 提出的无导方法解决这个问题；2）语言模型问题，本文通过对实验结果的观察改进了噪音通道模型，提出了 Bi-NC 模型；3）句子生成算法的问题， K&M 方法只是沿用文献[10]的算法，但该算法存在很多问题，本文提出一种自底向上的层级优化算法解决这个问题。后面三部分内容分别讨论以上三个问题。

3.2 无导的句子修剪方法

鉴于中文中并不存在大规模的原句-压缩句对齐语料，我们引入了 T&C 提出的无导句子修剪算法。这种算法的核心思想是：既然没有原句到修剪句的平行句对用来生成拓展概率，可以将整个语料库中所有的规则同时看成是原句和修剪句，用 rl 的频次占所有可以由 rs 拓展出来的规则的频次总和的比例来估计拓展概率，即：

$$P(rl\,|\,rs)=\frac{Count(rl)}{Count_{rl'\ s.t.\ rs\ svo\ rl'}(rl')}$$

其中 *rs svo rl'* 指规则 rs 可以拓展出（Shorter Version Of）规则 *rl'*，它们具有的特点是：*rs*、*rl'*左边相同，*rs* 的右边是 *rl'*的右边的子集。Count()是指特定规则的频次。

表 1 给出了规则 $IP\rightarrow PP\ PU\ NP\ VP\ PU$ 部分子规则以及它们的拓展概率。从表 1 可以看出，与源模型能够表示句法树合乎语法的程度不同，拓展概率不能直接衡量句法树的合理性，而只是给出 rs 有多少可能拓展出 rl。例如 $IP\rightarrow NP\ VP\ PU$ 应当是比较常见和合理的规则，但是拓展概率（0.092209）却远低于 $IP\rightarrow PP\ NP\ PU$ 这样的不常见规则(0.707492)。 因为 $IP\rightarrow NP\ VP\ PU$ 有可能拓展出许多其他规则，而 $IP\rightarrow PP\ NP\ PU$ 则相对而言没有太多选择。表 1 的例子还表明，无导的方法虽然没有使用对齐语料，仍然能够比较合理的表示规则之间的拓展概率。

3.3 Bi-NC 模型

T&C 不仅提出了无导句子修剪的方法，也指出了 K&M 方法存在的问题。T&C 指出，由于 K&M 方法在源

表 1 IP → PP PU NP VP PU的部分子规则及拓展概率

Subrules	Count(l)	Count(l')	拓展概率
IP → PP PU NP VP PU	12899	17933	0.719288
IP → PP PU NP PU	12899	18117	0.711983
IP → PP NP PU	12899	18232	0.707492
IP → PU VP	12899	85976	0.15003
IP → NP VP PU	12899	139889	0.092209
IP → VP	12899	1002094	0.012872

模型阶段同时考虑了两种不同的条件概率，导致给出的概率之和不为 1，因此不是一个完备的语言模型。改进的思路是，既然噪音通道模型与 bigram 是不同的模型，就应当分别进行归一化。由于改进后的语言模型分别独立的考虑了二元模型（Bigram Model）和噪音通道模型（Noisy Channel Model），因此被称为 Bi-NC 模型。

对于从同一个父节点出发的所有可能子树，计算它们的噪音通道模型概率和 bigram 概率，分别进行归一化，最后组合在一起，从而保证两个模型的独立性。即：

$$P_{Bi-NC} = P_{bigram}(s) \bullet P(s \mid l) = \frac{P_{bigram}(s)}{\sum_{S} P_{bigram}(s)} \bullet \frac{P(s)P(l \mid s)}{\sum_{S} P(s)P(l \mid s)}$$

改进以后，噪音通道模型与二元模型相互独立，同一个父节点出发的各子树两个模型的概率之和分别为 1，保证了模型本身的完备性。由于 Bi-NC 模型是两个独立模型的组合，在具体实验中可以测试每个模型单独使用的效果。

实验初期利用 Bi-NC 模型进行中文句子修剪的过程中，发现得到的最优修剪句都是很短的句子，初始实验效果较差。分析其中的原因发现，无论是计算 PCFG 概率还是 bigram 概率，如果按照通常的方法计算，都会倾向于获得较短的句子。K&M 同样在研究过程中发现了这个问题，他们无法直接生成最优修剪句，只好采用“长度选择”的方法（即给长度更长的句子更高的得分）来选择最优修剪句。而从理论上说，句子修剪的目的是获得更短的句子，给长句子更高的得分是不合理的。长度问题应该在模型建立阶段被解决。可行的解决方案是：改进通常计算 bigram 概率和 PCFG 概率的方法。通常词串 $w_{i-1}w_i$ 的 bigram 概率为：

$$P_{bigram}(w_i \mid w_{i-1}) = \frac{Count(w_{i-1}w_i)}{Count(w_i)}$$

这样的计算方法不利于长句，因为长句通常包含更多的词语，乘积会更小。改进的计算方法是：

$$P'_{bigram}(w_i \mid w_{i-1}) = Count(w_{i-1}w_i)$$

改进后句法树s 的 bigram概率为：

$$P'_{bigram}(s) = \prod_{i=2}^{Wordnum(s)} P'_{bigram}(w_i \mid w_{i-1}) = \prod_{i=2}^{Wordnum(s)} Count(w_{i-1}w_i)$$

其中 Wordnum(s)是 s 包含的词语数。通常重写规则 $A \to \beta$ 的 PCFG 概率由以下公式获得：

$$P_{cfg}(A \to \beta \mid A) = \frac{Count(A \to \beta)}{\sum_{\gamma} Count(A \to \gamma)}$$

其中 $\sum Count(A \to \gamma)$ 是所有由 A 出发的重写规则的总频次。但越是常用成分（例如 IP，NP，VP，NN，VV）越会在求 PCFG 概率时处在不利的位置，因为它们都要除以一个比不常用成分（例如 CC， DT，CD）更大的分母；而对于句子修剪来说，越是常用的成分越是需要留下来的成分，应该给常用的成分一个更高的概率。

改进的方法是：在计算 PCFG 概率的同时，不仅估计由父节点出发的某一规则的概率，也估计父节点自身的概率。即：

$$P'_{cfg}(A \to \beta \mid A) = P_{cfg}(A \to \beta \mid A) * P(A) = \frac{Count(A \to \beta)}{\sum_{\gamma} Count(A \to \gamma)} * \frac{\sum_{\gamma} Count(A \to \gamma)}{\sum_{\alpha}\sum_{\gamma} Count(\alpha \to \gamma)} = \frac{Count(A \to \beta)}{\sum_{\alpha}\sum_{\gamma} Count(\alpha \to \gamma)}$$

其中 $\sum_{\alpha}\sum_{\gamma} Count(\alpha \to \gamma)$ 是所有规则的总频次，对所有规则都来说都是相同的。因此：

$$P'_{cfg}(A \to \beta \mid A) = Count(A \to \beta)$$

改进后句法树s的PCFG概率为：$P'_{cfg}(s) = \prod_{rs \in s} P'_{cfg}(rs) = \prod_{rs \in s} Count(rs)$

其中 rs 是句法树 s 中包含的所有重写规则。

经过改进以后的 Bi-NC 模型由下式计算句法树 s 作为句法树 l 的修剪句的概率：

$$P_{Bi-NC} = P_{bigram}(s) \cdot P(s \mid l) = \frac{\prod_{i=2}^{Wordnum(s)} Count(w_{i-1}w_i) \prod_{rs \in s} Count(rs) \prod_{rs \in s, rl \in l, rs \to rl} \frac{Count(rl)}{Count_{rl' \in l, rs \to rl'}(rl')}}{\sum_{s} \prod_{i=2}^{Wordnum(s)} Count(w_{i-1}w_i) \sum_{s} \prod_{rs \in s} Count(rs) \prod_{rs \in s, rl \in l, rs \to rl} \frac{Count(rl)}{Count_{rl' \in l, rs \to rl'}(rl')}}$$

在计算过程中会出现 bigram 频次或重写规则频次为 0 的情况，都采取平滑处理，赋予很小但不为 0 的概率。改进以后的实验结果似乎对长句更为有利，但频次为 0 的情况可以过滤掉许多明显不合理的长句。

3.4 句子生成器和自底向上层级优化算法

理论上说，按照 Bi-NC 模型可以计算任何一个候选句的得分，并挑选出最优修剪句。而实际上，无论英文还是中文，一个长句都存在数以万计的候选句，逐一给出这些候选句的概率将需要很长的时间，无法满足实际应用的需求。因此必须采用一定的优化算法减少不必要的重复计算，以提高系统效率。

K&M 方法采用了文献[10]提出的句子生成优化算法。该算法的问题在于仅仅考虑了 n-gram 概率，而没有考虑句法分析问题。由于 K&M 需要考虑 bigram 概率和句法分析概率，如果直接使用（正如文中提到的那样）文献[10]的方法，只有两种可能的结果：1）过程未被优化；2）过程被优化，但舍弃的句子有可能是最优修剪句。也就是说，K&M 方法采用的优化算法不适合他们所使用的模型，采用这样的算法会使实验效果变差。我们在最初的实验中发现了同样的问题，并通过改进这种算法，提出了一种自底向上的层级优化算法。

优化的思路是：对于同一个父节点并且首尾词语相同的子树，只保留得分最高的。因为 1）只要父节点相同，各不同子树的 PCFG 概率和拓展概率都和父节点覆盖范围以外的计算独立；2）只要首尾词语相同，各个子树的 bigram 概率都和父节点覆盖范围以外的计算独立。两个条件一起使用，保证每一步优化操作都对父节点覆盖范围以外的操作没有影响。图 2 给出了自底向上的层级优化算法的伪程序代码。整个算法按照自底向上的原则，逐层处理各个节点，对每个节点的子树个数进行优化，直到根节点，最后排序输出修剪句。函数 Node()、Subtree()分别获得对应序号的节点和子树；函数 P()获得句法树的 Bi-NC 模型得分；HashMap0、HashMap1 沿用了 Java 中的 HashMap 数据结构。

图 3 显示了采用优化算法前后，处理不同长度的句子所需要的时间。横轴为被修剪句法树的节点数，在 30550 句实验语料中，平均每个节点约对应着 0.48 个汉字或标点符号。纵轴为系统生成结果所需要的时间，单位为秒。具体统计过程中以 5 个节点长度为间隔，随机抽取了覆盖各种长度的 300 个句子进行实验。实验所用计算机型号

```
For I=句法树的层数 to 1 {
 For J=1 to 句法树的节点数{
  If(Node[J]的层数==J && Node[J]不是叶子节点){
   For m=1 to Node[J]覆盖的句法树可能子树的个数{
    key=Subtree[m]的首尾词语;
     If(!HashMap0.contains(Key) ||
P(Subtree[m])>HashMap1.get(key)){
      HashMap1.put(key, P(Subtree[m]));
      HashMap0.put(key,m);}}
   用 HashMap0 的值对应的子树的集合作为 Node[J]所有
可能子树;
如果是根节点，排序输出;}}}
```

图2　自底向上层级优化算法

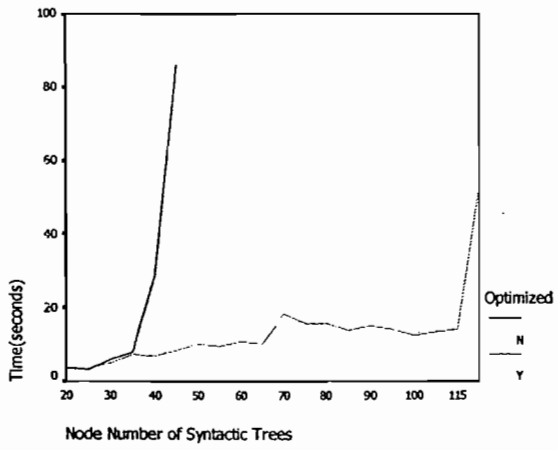

图3 优化算法对处理不同长度的句子所需时间的影响

为 Pentium IV/1024M。从图中可以看出，虽然优化算法处理范围还是有一定限制，但与不采用优化算法相比，无论处理范围和处理时间，优化算法的作用都非常显著。

4 实验结果与分析

中文句子修剪首先要解决训练语料问题。选取从五个门户网站（新浪、网易、腾讯、搜狐、Tom）上下载 2008 年 10 月 1 日至 2008 年 12 月 31 日的六个领域（国内社会、国际社会、体育、财经、科技、娱乐）的部分网页作为实验语料。预处理模块过滤了不完整的句子。为减少生成句法树所耗费的时间，过滤了 70 字及以上的句子。采用厦门大学史晓东教授开发的 SEGTAG 分词软件进行中文分词（不加词性标记），分词结果采用开源软件 Stanford Parser[11][12]中的中文句法分析工具[13]进行句法分析，最终生成了的 216208 句规模的中文树库。

通常从三个方面评价句子修剪的效果：1）合乎语法的程度；2）包含原句中重要信息的程度；3）句子的压缩率。具体评测过程中，首先随机抽取两组各 32 个句子，平均长度分别为 17.16 和 27.5 个汉字。为了避免由于分词和句法分析带来的错误对后面的实验造成的影响，一旦发现随机抽取的句子含有分词或句法分析错误，则重新抽取，直到获得每组 32 个句子为止。对于每组句子给出 6 个候选结果：1）Baseline，改进以前的噪音通道模型（即不修改 bigram 和 PCFG 的计算方法，不进行分别归一化），但采用无导的方法作为通道模型；2）Human，人工修剪句；3）System0，Bi-NC 模型；4）System1，Bi-NC 模型框架，但仅使用 bigram 模型；5）System2，Bi-NC 模型框架，但仅使用噪音通道模型；6）System3，Bi-NC 模型，但没有使用优化算法。表 2 显示了各系统的区别。

表 2 各系统使用方法的差别

	改进前的噪音通道模型	Bi-NC 模型		无导句子修剪方法	优化算法
		bigram 模型	改进后的噪音通道模型		
Baseline	√			√	√
System0		√	√	√	√
System1		√		√	√
System2			√	√	√
System3		√	√	√	

由 4 名评测者分别对两组结果从符合语法（Grammaticality）、包含重要内容（Importance）两个角度给出 1 到 5 的 5 级评分，得分越高表示越符合语法或内容越重要。评测者在评测时并不知道每个候选句来自哪个系统。表 3 给出了各系统句子压缩率（Compressed Rate），以及合乎语法程度(Grammaticality)、句子内容重要度（Importance）的平均得分和标准差。由于 System3 无法生成长句组的结果，因此该组只有 5 个候选结果。

表 3 的给出了评测结果。System0 在语法和重要度方面优于其他 5 个系统，但和人工方法还是有一定差异。综合考虑 System0、System1、System2 的表现，说明在 Bi-NC 模型框架下，同时考虑 bigram 模型和噪音通道模型

表 3 人工评测结果

Sentence Set	Candidate	Compression Rate	Grammaticality	Importance
Short	Baseline	57.67%	2.87±0.85	2.77±0.76
	Human	76.76%	4.97±0.18	4.63±0.55
	System0	71.77%	4.31±0.64	4.19±0.78
	System1	33.64%	2.68±1.25	2.87±1.31
	System2	62.96%	3.68±1.19	3.77±1.26
	System3	66.26%	3.74±1.09	3.87±1.20
Long	Baseline	54.06%	3.22±1.39	3.03±1.23
	Human	80.32%	4.97±0.18	4.63±0.49
	System0	66.57%	4.22±0.66	4.34±0.75
	System1	32.83%	2.59±1.07	2.40±0.91
	System2	59.69%	3.66±0.97	3.59±1.04

能取得更好的效果，而单独考虑噪音通道模型的效果优于单独考虑 bigram 模型。子树，但并不一定给出和不使用优化算法时完全相同的结果。实验结果说明使用优化算法的结果略好，其原因还有待进一步研究。System0 在短句组语法得分更高，但在长句组重要度得分更高，并未发现随长度增加效果明显下降的现象；System1、System2 在长句组的表现都有一定程度的下降。

5 结束语

本文提出了一种中文句子修剪方法。在具体探索过程中，我们构建了较大规模的中文句子修剪语料库和中文树库，为今后开展这方面的研究打下了基础；我们提出的 Bi-NC 模型、自底向上的层级优化算法，对包括英文在内的其他语言的相关研究同样具有借鉴意义。更为重要的是，本文提出的方法是建立在开放资源的基础上的，新闻语料、Stanford Parser、无导的方法，这些都是可以免费获得的或者可行的，本文的工作是可以被重复和改进的。

在今后的研究中，我们将进一步改进中文句子修剪方法，采用更精确的语言模型描述句子修剪问题。事实上，英文中已经证明存在多种方法能进一步提高句子修剪的效果，例如添加少量人工规则、考虑语义角色标注、采用更精确的机器学习算法等，这些工作都有待今后做进一步的研究。

参 考 文 献

[1] H. Jing and K. McKeown.. Cut and paste based text summarization. In: Proc of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000). Seattle, Washington., 2000: 178～185

[2] S. Blair-Goldensohn, D. Evans, V. Hatzivassiloglou, K. McKeown, A. Nenkova, R. Passonneau, B. Schiffman, A. Schlaikjer, A. Siddharthan, and S. Siegelman.. Columbia University at DUC 2004. In: Proc of the 2004 Document Understanding Conference (DUC 2004) at HLT/NAACL 2004. Boston, Massachusetts, 2004: 23～30

[3] J. Conroy, J. Schlesinger, and J. Goldstein. CLASSY query-based multi-document summarization. In: Proc of the 2005 Document Understanding Conference (DUC 2005) at NLT/EMNLP 2005. Vancouver, Canada, 2005

[4] J. Conroy, J. Schlesinger, D. O'Leary, and J. Goldstein. Back to basics: CLASSY 2006. In: Proc of the 2006 Document Understanding Conference (DUC 2006) at HLT/NAACL 2006. New York, 2006

[5] Jing, H. Sentence reduction for automatic text summarization. In: Proc of the 6th Applied Natural Language Processing Conference. Seattle, WA, USA, 2000: 310～315

[6] Knight, K., & Marcu, D. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. Artificial Intelligence, 2002, 139(1): 91～107.

[7] Hori, C., & Furui, S. Speech summarization: an approach through word extraction and a method for evaluation. IEICE Transactions on Information and Systems, 2004, E87-D(1): 15～25.

[8] Turner, J., & Charniak, E. Supervised and unsupervised learning for sentence compression. In: Proc of the 43rd Annual Meeting of the Association for Computational Linguistics. Ann Arbor, MI, USA, 2005: 290～297

[9] James Clarke and Mirella Lapata. Global Inference for Sentence Compression An Integer Linear Programming Approach. *Journal of Artificial Intelligence Research*, 2008, vol. 31: 399～429

[10] I. Langkilde. Forest-based statistical sentence generation. In: Proc of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics. Seattle, WA, 2000: 170～177

[11] Dan Klein and Christopher D. Manning. 2003. Fast Exact Inference with a Factored Model for Natural Language Parsing. In Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, MA: MIT Press, 3～10.

[12] Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In: Proc of the 41st Meeting of the Association for Computational Linguistics, 2003: 423～430

[13] Roger Levy and Christopher D. Manning. Is it harder to parse Chinese, or the Chinese Treebank?. In: Proc of the 41st Conference of the Association for Computational Linguistics (ACL 2003). Sapporo, Japan, 2003: 439-446