基于同步树序列替换文法的统计机器翻译模型

蒋宏飞[1] 李 生[1] 张 民[2] 赵铁军[1] 杨沐昀[1]

摘 要 基于短语的模型是目前发展相对成熟的一种统计机器翻译 (Statistical machine translation, SMT) 模型. 但基于短语的模型不包含任何结构信息, 因而缺乏有效的全局调序能力, 同时不能对非连续短语进行建模. 基于句法的模型因具有结构信息而具有解决以上问题的潜力, 因而越来越受到研究者们的重视. 然而现有的大多数基于句法的模型都因严格的句法限制而制约了模型的描述能力. 为突破这种限制并将基于短语的模型的优点融入到句法模型中, 本文提出一种基于同步树序列替换文法 (Synchronous tree sequence substitution grammar, STSSG) 的统计机器翻译模型. 在此模型中, 树序列被用作为基本的翻译单元. 在这种框架下, 不满足句法限制的翻译等价对和满足句法限制的翻译等价对都可以融入句法信息并被翻译模型所使用. 从而, 两种模型的优点均得到充分利用. 在 2005 年度美国国家标准与技术研究所 (NIST) 举办的机器翻译评比的中文翻译任务语料上的实验表明, 本文提出的模型显著地超过了两个基准系统: 基于短语的翻译系统 Moses 和一个基于严格树结构的句法翻译模型.

关键词 统计机器翻译, 句法限制, 同步文法, 同步树替换文法, 同步树序列替换文法
中图分类号 TP391

Synchronous Tree Sequence Substitution Grammar for Statistical Machine Translation

JIANG Hong-Fei[1] LI Sheng[1] ZHANG Min[2] ZHAO Tie-Jun[1] YANG Mu-Yun[1]

Abstract Phrase-based models are the state-of-the-art statistical machine translation models. However, they can not effectively handle global reordering and discontiguous phrases due to the lack of structural information. While syntax-based models have the potential to attack these problems, they suffer from the strictly syntactic constraints. To address these constraints and integrate the advantages of phrase-based models into syntax-based models, a synchronous tree sequence substitution grammar (STSSG) based statistical machine translation (SMT) model is presented in this paper. This novel model uses the tree sequence as the basic translation unit. Therefore, both the syntactic translation equivalences and the non-syntactic translation equivalences equipped with syntactic information can be utilized in the translation. Experimental results on the NIST 2005 Chinese-English machine translation data-set show that the proposed method achieves significant improvements over baseline methods including a phrasal model, Moses, and a tree-based syntax model.

Key words Statistical machine translation (SMT), syntactic constraint, synchronous grammar, synchronous tree substitution grammar, synchronous tree sequence substitution grammar (STSSG)

在当前的统计机器翻译 (Statistical machine translation, SMT) 领域, 基于短语的机器翻译模型均存在两个主要的缺陷: 1) 没有任何结构信息, 从而不具有全局调序能力; 2) 限制短语是连续的词汇串, 从而无法对非连续的翻译等价现象进行建模. 为解决以上问题, 研究者相继提出并尝试了多种基于句法的机器翻译模型[1−9]. 在利用句法结构对语言翻译进行建模时, 同样也会遇到很多问题. 下面我们仅对两个和本文研究内容密切相关的问题进行讨论.

首先, 不同语言间存在着结构分歧 (Structural divergence) 问题, 即不同语言间句法结构间存在大量非同构 (Non-isomorphic) 对应现象. 如图 2 (a) 中所示句对的句法树之间就是一种非同构对应现象. 结构分歧问题的主要原因在于不同语言在表达同一事物时在句法结构上的系统性差异以及灵活的意译现象等[7, 10−11]. 因此, 基于句法的机器翻译模型需具备对非同构句法结构对应进行建模的能力. 然而, 许多基于语言学的同步上下文无关文法 (Synchronous contex-free grammar, SCFG) 的句法模型并不具备这种能力. 因为 SCFG 文法规则只允许同层次兄弟节点之间进行调序, 导致只能对同构的句法结构进行建模.

同步树替换文法 (Synchronous tree substitution grammar, STSG) 的规则允许不同层次的节

收稿日期 2008-05-13 收修改稿日期 2009-01-21
Received May 13, 2008; in revised form January 21, 2009
国家自然科学基金重点项目 (60736014) 和国家高技术研究发展计划 (863 计划) 重点项目 (2006AA010108) 资助
Supported by the Key Program of National Natural Science Foundation of China (60736014) and the Key Project of National High Technology Research and Development Program of China (863 Program) (2006AA010108)
1. 哈尔滨工业大学计算机科学与技术学院机器智能与翻译研究室 哈尔滨 150001 2. 新加坡信息通讯研究所 新加坡 119613
1. Machine Intelligence and Translation Laboratory, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P. R. China 2. Institute for Infocomm Research, Singapore 119613, Singapore
DOI: 10.3724/SP.J.1004.2009.01317

点进行调序, 从而具备对非同构结构的对应进行建模的潜力. Eisner 在文献 [8] 对利用 STSG 在依存句法结构上进行非同构结构对应的学习提出了一些构想. 蒋宏飞等在文献 [4] 中给出一种基于 STSG 文法的在短语结构句法上的模型, 并给出了规则抽取和翻译解码的过程描述.

另一个重要的问题就是句法限制的问题. 句法限制指翻译规则的源语言部分和目的语言部分都必须为句法树中的一个元树. 元树是一颗子树, 但其叶子节点可以是非终结符[4]. 在现有的一般句法模型中, 往往因严格的句法限制导致大量不满足句法限制的翻译等价对 (Non-syntactic translation equivalence) 不能得以利用. 而基于短语的机器翻译模型因无句法限制, 从而可以利用这些知识, 而且前人研究证实此类知识对翻译性能有很大帮助[12].

正如文献 [13] 所指出的, 基于短语的模型和基于句法的模型存在很大优势互补的空间. 一般的基于短语模型不具有任何句法信息, 不具备全局调序能力的同时也无句法限制; 而一般基于句法的模型具有结构信息, 可进行全局调序, 但严格的句法限制将浪费大量翻译知识. 鉴于此, 一个自然的想法就是: 是否可以将两者的优势进行融合? 在目前已有的研究基础上, 可能的融合途径有两种: 1) 将句法信息加入到短语模型中, 进而指导调序过程; 2) 对现有句法模型进行扩展和泛化, 使之可以利用不满足句法限制的翻译等价对. 前者在文献 [14] 中进行了尝试, 并取得了较好的效果. 文献 [5] 在文献 [3] 所提出的树到串模型基础上, 提出了一种通过森林到串规则来利用非句法翻译等价对增强原有模型能力的策略. 实验表明, 此策略显著提升了原有树到串模型的性能. 本文同样尝试从第二个方向进行探索, 期望将基于 STSG 文法的句法模型进行扩展和泛化, 使之可以利用不满足句法限制的翻译等价对. 本文所提模型和文献 [5] 所采用策略有以下两点关键区别. 首先, 文献 [5] 在使用森林到串规则时需要借助于辅助规则. 而辅助规则是在解码时动态构造生成的, 而不是从训练数据中学习得到的, 故无法进行概率估计. 而本文模型将树序列作为基本的翻译单元, 在解码过程中直接进行树序列到树序列的转化, 而且所使用的全部规则都是从训练语料中学习得到. 其次, 文献 [5] 所基于的模型仅考虑了源语言端的句法信息, 而本文模型同时考虑了源语言端和目的语言端的句法信息.

另外, 一些研究者也在研究其他技术对机器翻译的影响, 如文献 [15] 研究中文全词消歧对机器翻译系统的影响.

本文提出的基于同步树序列替换文法 (Synchronous tree sequence substitution grammar, STSSG) 的翻译模型是对文献 [4] 所提同步树替换文法模型的扩展和泛化. 与基于短语模型和基于词模型之间的关系类似, 本文所提模型中把基本翻译单元由树泛化为树序列 (1.1 节给出了树序列的定义). 图 1 演示了一个基本翻译单元的形象化的演变过程. 起初, 统计机器翻译模型中的基本翻译单元只是单个词 (Word); 后来, 演变为短语 (Phrase); 再后来, 出现了基于句法的模型, 将树作为基本翻译单元; 本研究中则将树序列作为基本翻译单元. 图 2 给出一个具体例子来对基于短语模型、基于同步树替换文法模型和基于同步树序列替换文法模型进行一个直观的比较. 从图 2 (a) 所示的带词对齐信息的双语句法树对中, 可以抽取出短语翻译等价对: (*把 钢笔 给*↔ *Give the pen*). 这个翻译等价中实际上模拟了一个动词谓语 (*给 / Give*) 和名词宾语 (*钢笔 / pen*) 之间的调序. 尽管如此, 因为没有泛化能力, 这个调序规则在实际翻译过程中的用处非常有限. 再来看一般的基于句法的模型. 因为规则 (*把 钢笔 给*↔ *Give the pen*) 两边的部分都不能被一个完整的子树恰好覆盖, 所以在一般基于句法的模型中, 类似的规则抽取不出来. 然而, 在本文所提出的基于同步树序列替换文法的翻译系统中, 这种翻译现象可以被学习出来. 比如可以抽取出规则: $(BA(把)\ NN(钢笔)\ VV(给) \leftrightarrow VB(Give)\ NP(DT(the)\ NN(pen)))$. 而且, 还可以通过将叶子节点中的词汇化节点进行泛化形成抽象规则来增强规则的扩展性. 图 2 (c) 给出了两个相关的规则示例. 从这个简单的例子中我们可以看出基于同步树序列替换文法模型的两个优势: 首先, 带有句法信息的不满足句法限制的翻译等价对也可以被模型使用; 其次, 可以继承并增强一般句法模型的全局调序能力. 本文对文献 [9] 进行了大量充实和完善. 本文着重强调了所提模型是一个树序列到树序列的翻译模

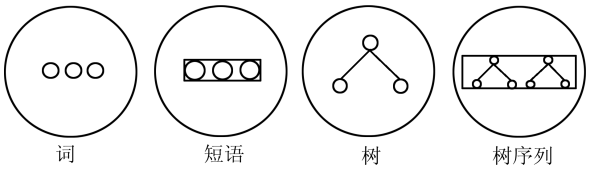

图 1 统计机器翻译中基本翻译单元 (Basic translation unit, BTU) 的演变 (最初的 BTU 是词; 短语是目前很多基于短语 SMT 模型的 BTU; 树是目前大多数基于句法的 SMT 模型的 BTU. 本研究中将树序列作为 BTU.)

Fig. 1 The evolution of BTU (Basic translation unit) in statistical machine translation (SMT) (Word is the original BTU. Phrase is a replacement of a word, and it is the BTU in many state-of-the-art phrasal systems. Tree is used as BTU in tree-based models. This study uses tree sequence as the BTU.)

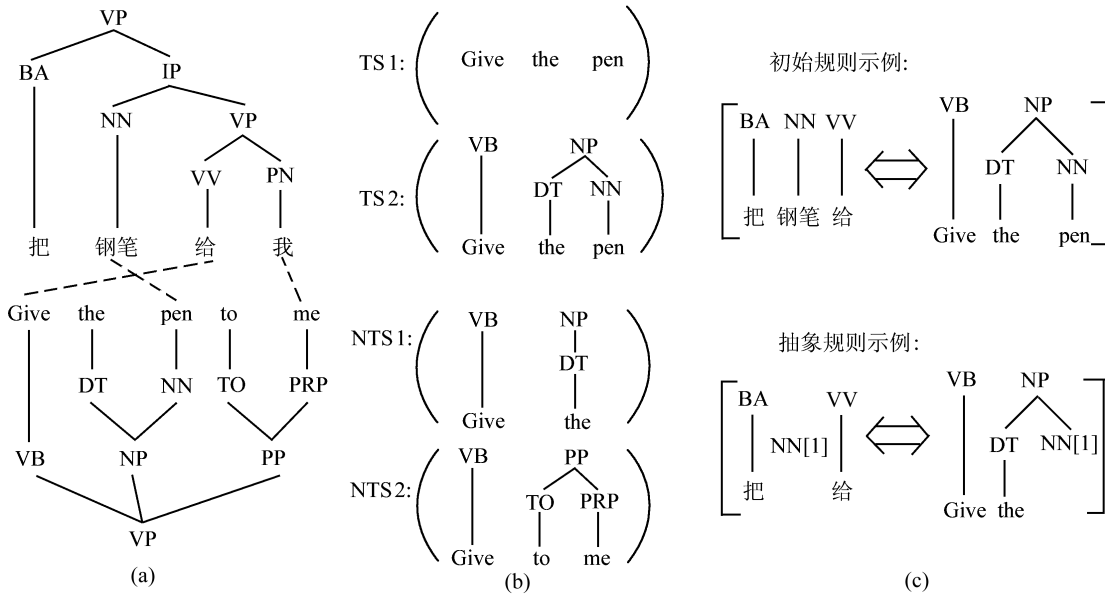

图 2　(a) 句法树对样例, (b) 相关的树序列正反例, (c) 所能抽取出来的规则示例

Fig. 2　(a) A syntax pair example, (b) The positive/negative example of tree sequence defined in this paper, (c) The rule examples that can be extracted from this syntax pair

型, 即本模型并不严格要求解码过程的输入端和输出端必须为一个完整的句法树, 只要是树序列即可. 此外, 本文给出了严格的树序列定义并给出了树序列获取的算法. 并且本文的数学模型建立在同步树序列替换文法的框架下 (文献 [9] 中, 数学模型是建立在树对齐模板上), 在模型的推导上对树序列到树序列的转化进行了较严格的形式化刻画.

下文如下安排: 第 1 节形式化地给出基于同步树序列替换文法的统计机器翻译模型; 第 2 节介绍规则抽取方法以及参数训练的一些问题; 第 3 节给出了解码过程; 第 4 节给出了实验设置以及实验结果; 最后, 第 5 节给出本文的结论.

1　基于同步树序列替换文法的机器翻译模型

1.1　树序列的定义

为避免混淆, 给出本文模型之前首先对本文所涉及的树序列给出一个形式化定义. 给定一个句子 w_1^n, 跨度(Span)$s=[i,j]$ 和其中的 w_i^j $(1 \le i \le j \le n)$ 对应. 给定句子的一个句法树 $T(w_1^n)$, 那么一个覆盖跨度 s_i^j 的树序列可以是:

1) $T(w_1^n)$ 的一个子树 t, 如果 t 的跨度正好为 w_i^j;

2) 一个有序的子树序列 $\{t_1, \cdots, t_k\}$, 如果下列条件得以满足: 假设 s_l 是子树 t_l 对应的跨度,

a) 子树完整性: $\forall\, l\,(1 \le l \le k)$, t_l 是 $T(w_1^n)$ 的一个完整子树;

b) 子树连续性: $\forall\, l\,(1 \le l < k), s_l = [m,n] \Rightarrow s_{l+1} = [n+1, x]$, $x \ge n+1$;

c) 子树不重叠性: $\forall\, x, y\,(1 \le x \ne y \le k) \Rightarrow span\{t_x\} \cap span\{t_y\} = \varnothing$.

3) 此外, 不是树序列.

在极端的情况下, 一个单词 w_i 可以被看成一个最小的树. 由此, 基于短语的 SMT 模型中的短语便可以被看作是树序列. 图 2 (b) 演示了树序列 (Tree sequence, TS) 和非树序列 (Non-tree sequence, NTS) 的例子.

1.2　同步树序列替换文法

STSSG 文法是同步树替换文法的一个扩展. 一个 STSSG 是一个七元组 $G = \langle \sum_f, \sum_e, N_f, N_e, P, S_f, S_e \rangle$, 其中:

1) $\sum_f$ 和 $\sum_e$ 分别代表源语言端和目的语言端终结符 (在本研究中指单词) 词表;

2) N_f 和 N_e 分别代表源语言端和目的语言端非终结符 (在本研究中指句法标记) 词表;

3) $S_f \in \sum_f$ 以及 $S_e \in \sum_e$ 分别代表源语言端和目的语言端起始符号;

4) P 代表产生式文法规则集合.

STSSG 文法中的每个文法规则是一个存在对齐关系的树序列对:

$$\langle \xi_f, \xi_e, \sim \rangle$$

其中 ξ_f 和 ξ_e 分别代表源语言端和目的语言端的树序列, $\sim$ 是一个一一对应关系集合. 其中的一一对应指 ξ_f 中的替换节点和 ξ_e 中的替换节点之间的对

应. 树序列中的非终结符叶子节点被称为是替换节点 (Substitution node).

1.3 基于同步树序列替换文法的翻译模型

一个翻译系统的基本任务就是将给定的源语言句子 f 翻译成一个合适的目的语言句子 e. 那么, 翻译模型就是要对 $Pr(e|f)$ 进行建模. 接下来, 我们给出基于 STSSG 文法的翻译模型数学建模过程.

给定一个源语言句子 f 和一个目的语言句子 e, 我们首先引入两个隐变量 $\Phi(f)$ 和 $\Phi(e)$, 两者分别代表覆盖 f 和 e 的树序列.

$$Pr(e|f) = \sum_{\langle \Phi(f), \Phi(e) \rangle} Pr(e, \Phi(e), \Phi(f)|f) \quad (1)$$

然后, $Pr(e, \Phi(f), \Phi(e)|f)$ 可以被进一步分解为三个子模型, 如式 (2) 所示.

$$\begin{aligned} Pr(e, \Phi(f), \Phi(e)|f) = {} & Pr(\Phi(f)|f) \times \\ & Pr(\Phi(e)|\Phi(f), f) \times \\ & Pr(e|\Phi(e), \Phi(f), f) \end{aligned} \quad (2)$$

其中, $Pr(\Phi(f)|f)$ 是一个树序列标注模型. 此模型对 f 所对应的所有可能树序列覆盖进行优化选择. 因为本文将基于同步树序列替换文法的翻译模型建立在一个对数线性模型框架上[16], 故而本文采用一个表示结构信息丰富程度的特征函数来对此模型进行建模:

$$\tau(\Phi(f), f, T(f)) = \sum_{n \in \Phi(f), m \in T(f)} (\boldsymbol{\delta}(n, m))$$

$\Phi(f)$ 代表覆盖 f 的一个树序列, $T(f)$ 代表 f 的句法树, n, m 分别代表 $\Phi(f)$ 和 $T(f)$ 中的任一节点. $\boldsymbol{\delta}(n, m)$ 是一个二值函数. 此函数在 n 和 m 相同时值为 1, 否则值为 0. 在目前的实现中, $\boldsymbol{\tau}(\Phi(f), f, T(f))$ 返回树序列 $\Phi(f)$ 中包含的节点数.

$Pr(e|\Phi(f), \Phi(e), f)$ 是一个译文生成模型. 因为在这个模型中, 树序列 $\Phi(e)$ 中的叶子节点 (词) 只需要按照从左到右的顺序依次读出然后拼接为最终的译文, 所以本模型分数恒为 1, 可以忽略.

最重要的子模型是 $Pr(\Phi(e)|\Phi(f), f)$. 它是源语言端树序列到目的语言端树序列的转换模型. 因为模型是基于同步树序列替换文法的, 所以可以把源语言端树序列到目的语言端树序列的转换过程作为同步树序列替换文法的一个推导 (Derivation) 过程来进行研究.

类似文献 [2] 的做法, 设 D 是文法的一个推导, 并设 $f(D)$ 和 $e(D)$ 是推导 D 产生的源语言端和目的语言端的字符串. 推导 D 可以被表示为三元组 $\langle r, i, j \rangle$ 的一个序列. 每个三元组代表一次文法规则的使用: 用规则 r 来对源语言端的跨度 $f(D)_i^j$ 进行重写. 那么, $Pr(\Phi(e)|\Phi(f), f)$ 就可用推导 D 的得分进行模拟. 而 D 的得分可以由其使用的规则的乘积进行计算.

$$w(D) = \prod_{<\langle r,i,j \rangle \in D} w(r) \quad (3)$$

每个独立规则的分数是其对应的特征函数在对数线性框架下的综合得分:

$$w(r: \langle \xi_f, \xi_e, \sim \rangle) = \prod_i \Phi_i(r: \langle \xi_f, \xi_e, \sim \rangle)^{\lambda_i} \quad (4)$$

其中, Φ_i 是定义在规则上的特征函数. 在本研究中, 整个翻译模型采用了下列特征:

1) $P(\xi_f|\xi_e)$ 和 $P(\xi_e|\xi_f)$, 双向的树序列映射概率;

2) $lex(\xi_f|\xi_e)$ 和 $lex(\xi_e|\xi_f)$, 双向的词汇化权重;

3) $P(e)$, 译文语言模型得分;

4) L, 译文词数, 即词惩罚;

5) N 翻译过程 (文法推导过程) 所用到规则个数, 也即规则数惩罚;

6) $\boldsymbol{\tau}(\xi_f)$, 源语言端结构信息丰富程度表征特征.

2 基于同步树序列替换文法机器翻译模型的训练

2.1 树序列的获取

在本文的实现中, 对于给定句子 w_I^J 的任一跨度, 其对应的所有树序列都是从句法树 $T(w_I^J)$ 中分解出来的. 树序列获取的伪代码在算法 1 中进行了描述. 在 1∼3 行中, 映射 Λ(键是一个跨度, 值是相对应的树序列集合) 用 $T(w_I^J)$ 中所有的 ⟨树节点对应跨度, 树节点⟩ 对进行初始化. 算法 1 的剩余部分逐步对每个跨度对应的树序列集合进行扩展. 扩展的途径类似于一个动态规划过程, 通过不断把子跨度划分对应的树序列集合进行合并来获得新的树序列集合. 算法 1 第 6 行中的 merge 操作负责通过计算 $\Lambda[(i,k)]$ 和 $\Lambda[(k+1,j)]$ 的笛卡尔乘积来为 $\Lambda[(i,j)]$ 获取新的树序列.

算法 1. 树序列获取算法

输入: 句法树 T

1: **for** T 中的任一节点 m 及其对应跨度 s **do**
2: 将 $\{m\}$ 加入 $\Lambda[s]$
3: **end for**
// u 是当前跨度大小
// N 是句法树 T 对应句子所含词个数
4: **for** $u \leftarrow 1, \cdots, N$ **do**

5: **for each** i, j, k s.t. $u = j - i$ and $1 \le i < j \le n, i < k < j$ **do**
6: $\Theta = \text{merge}(\Lambda[(i,k)], \Lambda[(k+1,j)])$
7: 将 Θ 加入 $\Lambda[(i,j)]$
8: **end for**
9: **end for**
输出: 一个跨度与树序列的对应结构 Λ.

2.2 基于同步树序列替换文法的机器翻译模型的规则自动抽取

在本文的实现中, 树序列到树序列的翻译转化规则 (即同步树序列替换文法的产生式规则) 是从词对齐语料: 三元组 $\langle T(f), T(e), \sim \rangle$ 的集合中抽取出来的. 其中, $T(f)$ 是源语言句子 f 的句法树, $T(e)$ 是目的语言句子 e 的句法树, $\sim$ 是一个 f 和 e 中词之间的多对多映射关系.

为下文论述方便, 先给出对应跨度对的定义.

对应跨度对 (Corresponding span pair, CSP). 和文献 [2] 中的初始短语对 (Initial phrase pair) 的定义类似, 给定一个词对齐句对 $\langle f, e, \sim \rangle$, 一个跨度对 $\langle s(f_i^j), s(e_{i'}^{j'}) \rangle$ 是一个对应跨度对, 当且仅当:
1) $f_k \sim e_{k'}$ for 某下标 $k \in [i,j]$ 并且 $k' \in [i',j']$;
2) $f_k \nsim e_{k'}$ for 某下标 $k \in [i,j]$ 并且 $k' \notin [i',j']$;
3) $f_k \nsim e_{k'}$ for 某下标 $k \notin [i,j]$ 并且 $k' \in [i',j']$.

在真正的抽取动作开始前, 首先需要利用算法 1 给出的树序列获取算法为源语言句子和目的语言句子的每个跨度进行树序列的获取.

本文将所要抽取的规则分为两类并进行分步式抽取. 第一类称为初始规则 (Initial rule), 第二类称为抽象规则 (Abstract rule). 初始规则指叶子节点都是终结符 (单词) 的规则, 抽象规则指叶子节点中包含至少一个非终结符 (句法标记) 的规则. 图 2 (c) 中给出了初始规则和抽象规则的示例.

初始规则的抽取比较直接. 抽取过程只需依次遍历每个跨度对 (s_f, s_e), 并判断其是否为 CSP, 如果是, 记录下来; 否则, 跳过. 然后, 对于每个 CSP(s_f, s_e), 源语言端跨度对应的树序列集合 $\Phi(s_f)$ 和目的语言端跨度对应的树序列集合 $\Phi(s_e)$ 进行笛卡尔乘积 ($\Phi(s_f) \bigotimes \Phi(s_e)$), 形成初始规则. 抽象规则的抽取则相对复杂, 抽象规则都是在初始规则的基础上衍生出来的. 算法 2 给出了抽象规则的抽取过程.

算法起始时 (行 1), 用 $\Re_{\text{ini}}$ (已抽取得到的初始规则集合) 初始化 $\Re_{\text{all}}$ (装载全部规则的队列). 接下来, 算法依次扫描队列 $\Re_{\text{all}}$ 的下一个元素, 即规则, 如 $(r = \langle \xi_f, \xi_e, \sim \rangle)$, 并考察其是否可以被进一步抽象化 (行 5 ∼ 9). 函数 abstract$((s_i^j, s_{i'}^{j'}), r)$ 通过将规则 r 中覆盖源语言端跨度 s_i^j 和目的语言端跨度 $s_{i'}^{j'}$ 的部分同时进行规约替换为一个规则 r 中没使用过的编号而形成新的抽象规则. 产生的新抽象规则又被加入到 $\Re_{\text{all}}$ 的尾部 (行 11). 当算法扫描到 $\Re_{\text{all}}$ 队尾时整个过程结束, 并将整个规则集合 (包含初始规则和抽象规则) 输出 (行 3). 图 2 (c) 所示的抽象规则就是在上面的初始规则基础上将 CSP([钢笔], [pen]) 规约后形成的. 图 3 给出了更多可以从图 2 (a) 所示的句对中抽取出来的规则的示例.

算法 2. 抽象规则抽取算法

输入: 初始规则集合 $\Re_{\text{ini}}$, 句法树对 $\langle T(f), T(e), \sim \rangle$
1: $\Re_{\text{all}} = \Re_{\text{ini}}$;
// 将 $\Re_{\text{all}}$ 看做队列进行操作
2: **if** Rear$(\Re_{\text{all}})$ = Head$(\Re_{\text{all}})$ **then**
3: **return** $\Re_{\text{all}}$
4: **end if**
5: $(r = \langle \xi_f, \xi_e, \sim \rangle)$ = Head$(\Re_{\text{all}})$;
6: Head$(\Re_{\text{all}})$ = Next$(\Re_{\text{all}})$;
7: **for each** 被 ξ_f 覆盖的子跨度 s_i^j **do**
8: **for each** 被 ξ_e 覆盖的子跨度 $s_{i'}^{j'}$ **do**
9: **if** $(s_i^j, s_{i'}^{j'})$ 是一个 CSP 并且在 r 中尚未被抽象 **then**
10: Θ = abstract$((s_i^j, s_{i'}^{j'}), r)$;
11: 将 Θ 加入 $\Re_{\text{all}}$ 的队尾;
12: **end if**
13: **end for**
14: **end for**
15: goto step 2;
输出: 规则集 $\Re_{\text{all}}$.

另外, 在对未对齐词的处理上, 本文模型为了能够覆盖所有短语, 采取了和短语模型 (如 Moses) 类似的策略. 例如, 图 2 (a) 中的 “我” 可以对应 “me”, 也可以对应 “to me”; “钢笔” 可以对应 “pen”, 也可以对应 “the pen”.

2.3 模型复杂度控制策略以及基于 STSSG 的模型和其他模型之间的关系

在实际操作中, 前文所描述的规则抽取算法将产生数量非常庞大的规则集合, 如果不加以控制, 将会使得模型训练、模型解码在时间空间消耗上变得不可接受. 为了能对整个模型的复杂度进行有效控制, 本文采用了以下几个控制策略:

1) 一个树序列所包含的树个数不能多于 α;
2) 一个规则中包含的抽象节点个数不能超过 β;
3) 一个规则中树高不超过 γ;
4) 每个跨度对应的抽象规则数不能超过 ω;
5) 每个跨度对应的树序列个数不超过 π.

为了对每个规则的翻译概率进行估计, 类似文献 [2] 的做法, 本文实现中首先对每个初始规则分配一定的计数值 (比如 1), 然后把这个计数值平均分配到从此初始规则衍生出的抽象规则上. 然后基于这种计数分配策略, 再采用最大似然估计(Maximal

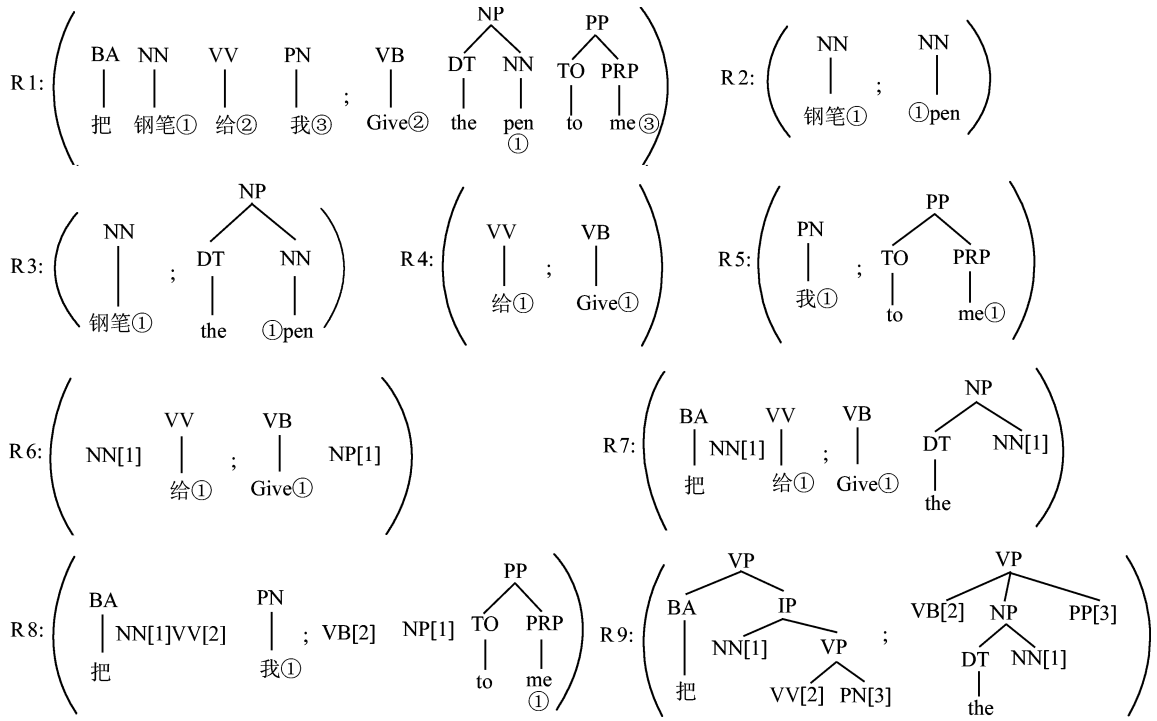

图 3 从图 2 (a) 中抽取出的规则示例 (圆圈内同样标号代表终结符之间的对应关系, 方括弧内的同样标号代表非终结符之间的对应关系.) R 1 ∼ R 5 是初始规则, R 6 ∼ R 9 是抽象规则; R 2, R 3, R 4, R 5, R 9 是树到树转换规则, R 1, R 6, R 7, R 8 是树序列到树序列 (树个数 ≥ 2) 转换规则.)

Fig. 3 Some rules extracted from the sentence pairs illustrated in Fig. 2 (left part) (Cycled indexes represent the correspondences between terminals from source and target side, square bracketed indexes represent the correspondences between non-terminals from source and target side. R 1 ∼ R 5 are initial rules, R 6 ∼ R 9 are abstract rules; R 2, R 3, R 4, R 5, R 9 are tree to tree transformation rules, R 1, R 6, R 7, R 8 are tree sequence to tree sequence (tree number ≥ 2) transformation rules.)

likelihood estimation, MLE) 来对 $P(\xi_f|\xi_e)$ 和 $P(\xi_e|\xi_f)$ 进行计算. 对数线性模型中的参数权重由最小错误率训练进行估计[17].

有趣的是, 如果对模型控制参数进行相应的设置 (如 1323 页式 (5)), 本文提出模型就可以变成其他模型, 如基于 STSG 的模型、基于 SCFG 的模型以及单调的基于短语的模型. 即本文所提模型可以看成是其他模型的一个一般化形式.

算法 3. 基于 STSSG 文法模型的解码算法

输入: **规则集** R, **源语言端输入句法树** $T(f)$

$\Phi(f)$ = TreeSequenceAcquisition($T(f)$);
O = GetTransOption($\Phi(f)$,R);
// u 是当前跨度大小
// N 是句法树 $T(f)$ 对应句子所含词个数
for $u \leftarrow 1, \cdots, n$ **do**
  for $i \leftarrow 1, \cdots, n$, $j = i + u - 1$ **do**
    for each $O[i, j]$ 中的翻译候选项 p **do**
      if p 是一个不包含替换节点的翻译候选 **then**
        将 p 加入 $H[i, j]$;
      else
        Θ = Substitute(p,H);
        将 Θ 加入 $H[i, j]$;
      end if
    end for
  end for
end for

输出: 目的语言译文 e

3 基于 STSSG 的机器翻译模型的解码

大多数的基于短语的机器翻译系统均采用一个从左到右的线性解码过程. 而大多数基于树结构的句法机器翻译系统[3−4] 采用一种自底向上的依次处理各个树节点的策略. 然而, 这两种解码流程对本文所提出的基于 STSSG 模型的解码都不适用.

本文采用的解码过程类似 CKY (Cocke-Kasami-Younge) 句法分析过程[2]. 从实现上讲, 本算法是一个基于集束搜索 (Beam search) 的栈解码过程. 需要指出的是, 本算法在安排栈结构的时候是以每个跨度对应一个栈的策略进行的. 算法按照从左到右、从小到大的顺序对输入的源语言句子 f 的每个跨度依次进行处理. 在搜索步骤中, 所有对应于

同一个源语言端跨度的中间翻译结果 (Hypotheses) 被存在同一个栈中. 包含一个词的跨度最先被处理, 然后依次处理更大的跨度, 当处理一个大的跨度时, 其子跨度的所有已获得的译文便可以被利用. 最终, 翻译完最大跨度, 即 $[1, n]$(n 为源语言句子长度), 算法终止.

算法 3 给出了解码过程的伪代码. 其中第 9 行中的 Substitute 操作就代表用子跨度已获得的译文跨度中替换节点进行翻译的过程. 从文法角度来看, 基于 STSSG 的机器翻译模型的解码过程就是一个 STSSG 文法推导. 图 4 给出了两个文法推导的示例, 这些推导对应利用表 1 中给出的文法规则对图 2 (a) 给出的中文句子的翻译过程. 推导 1 所用规则都是一般的 STSG 文法的树到树转化规则, 推导 2 则利用了树序列到树序列的转化规则. 从这个例子也可以看出, 在基于 STSSG 的模型的推导中包含基于 STSG 文法的模型的推导. 值得注意的是, 从推导 2 可以看出, 本文模型在最终翻译完成时, 对源语言句子和目的语言句子形成的句法覆盖并不一定必须是一个完整的句法树, 而可以是任何树序列.

现实中, 为控制解码算法的搜索复杂度, 需要用到一些阈值参数进行控制. 本文实现中, 采用了栈规模限制 b 和翻译候选个数限制 a (用来限制从翻译模型中载入的对应于同一源语言端树序列的翻译候选个数, 类似于短语模型系统, 例如 Moses 中的 "-ttable-limit"), 以及和规则抽取部分类似的参数: α, β, γ, π.

4 实验验证及讨论

4.1 实验设置

依照本文提出的基于 STSSG 的机器翻译模型, 我们实现了一个原型系统, 为论述方便将其命名为 Lanchier.

为验证本文所提模型相对于基于短语的模型和一般的基于句法的模型的优越性, 选择目前最为流行的基于短语的系统 Moses[18] 和一个基于 STSG 文法的句法系统 (通过将 STSSG 系统中参数 α 设置为 1 进行实现) 作为基准系统.

Moses 的参数设置. 实验中采用了 Moses 的默认设置. 在其中的调序模型选项中, 选用了 "msd-bidirectional-fe".

Lanchier 的参数设置. 实验中对 Lanchier 采用了如下设置: $\alpha = 4$ (STSG 系统中, 此参数为 1), $\beta = 6$, $\gamma = 6$, $\omega = 50$, $\pi = 5$, $a = 20$, $b = 50$.

$$\text{STSSG 模型} \Rightarrow \begin{cases} 1)\ \text{单调的短语模型,} \quad \text{设置 } \beta = 0\ ,\ \gamma = 1 \\ 2)\ \text{基于同步上下文无关文法的模型,} \quad \text{设置 } \alpha = 1,\ \gamma = 2 \\ 3)\ \text{基于同步树替换文法的模型,} \quad \text{设置 } \alpha = 1 \end{cases} \tag{5}$$

推导 1:

$$\begin{aligned} \text{Start} &\xrightarrow{\text{R 9}} \text{VP(BA(把)IP(NN[1]VP(VV[2]PN[3]))) } \leftrightarrow \text{ VP(VB[2]NP(DT(the)NN[1])PP[3])} \\ &\xrightarrow{\text{R 2}} \text{VP(BA(把)IP(NN(钢笔)VP(VV[2]PN[3]))) } \leftrightarrow \text{ VP(VB[2]NP(DT(the)NN(pen))PP[3])} \\ &\xrightarrow{\text{R 4}} \text{VP(BA(把)IP(NN(钢笔)VP(VV(给)PN[3]))) } \leftrightarrow \text{ VP(VB(Give)NP(DT(the)NN(pen))PP[3])} \\ &\xrightarrow{\text{R 5}} \text{VP(BA(把)IP(NN(钢笔)VP(VV(给)PN(我))))}\leftrightarrow \\ &\quad \text{VP(VB(Give)NP(DT(the)NN(pen))PP(TO(to)PRP(me)))} \end{aligned}$$

推导 2:

$$\begin{aligned} \text{Start} &\xrightarrow{\text{R 8}} \text{BA(把) NN[1] VV[2] PN(我) } \leftrightarrow \text{ VB[2] NP[1] PP(TO(to)PRP(me))} \\ &\xrightarrow{\text{R 3}} \text{BA(把) NN(钢笔) VV[2] PN(我) } \leftrightarrow \text{ VB[2] NP(DT(the)NN(pen)) PP(TO(to)PRP(me))} \\ &\xrightarrow{\text{R 4}} \underline{\text{BA(把)}}\ \underline{\text{NN(钢笔)}}\ \underline{\text{VV(给)}}\ \underline{\text{PN(我)}} \leftrightarrow \\ &\quad \underline{\text{VB(Give)}}\ \underline{\text{NP(DT(the)NN(pen))}}\ \underline{\text{PP(TO(to)PRP(me))}} \end{aligned}$$

图 4 用图 3 中所列规则对图 2 (a) 所示句法树对进行文法推导的示例 (推导 1 是一个基于 STSG 文法的推导, 而推导 2 是一个基于 STSSG 文法的推导.)

Fig. 4 Two derivations for translating the Chinese sentence in Fig. 2 (left part) using the rules in Fig. 3 (Derivation 1 is an STSG derivation and Derivation 2 is an STSSG derivation.)

实验采用的双语平行训练数据是 FBIS (Forgein broadcast information service) 数据集 (该语料含汉语单词 7.06 M, 英语单词 9.15 M). 多对多的词对齐信息通过在训练数据上运行双向的 GIZA++ 工具[19], 然后采用 "grow-diag-final" 启发式规则获得. 我们采用 SRI 的语言模型工具包在 Gigaword 语料的新华部分 (约 181 M 英语词) 上训练得到一个 4 元的语言模型. 平滑策略采用修正的 Kneser-Ney 方法. 实验所用开发集包含 486 个句子, 均来自 2002 年 NIST 机器翻译评测中中英翻译任务中的测试集. 实验测试集是 2005 年 NIST 机器翻译评测中中英翻译任务中的测试集[1], 共包含 1 082 个句子.

训练语料中的中文句子、英语句子以及开发集和测试集中的中文句子都需要进行句法分析. 我们采用 Stanford 句法分析工具包中的中英文工具分别对中文和英文句子进行了自动句法分析. 评测工具采用 NIST 官方实现的 Bleu 指标计算脚本 (版本为 11a). 除了采用了大小写敏感的 ngram 匹配设置外, 其他设置均默认. 显著性测试采用了 Zhang 等实现的自举重采样策略[20].

4.2 规则统计分析

基于 4.1 节给出的设置, 我们在 FBIS 数据集上进行了规则的抽取, 并用测试集对抽取出的规则进行了过滤. 同时, 我们对 Moses 所抽取的规则也做了同样的处理. 表 1 给出了整体模型中包含的规则集的统计结果. 表 2 给出了用测试集过滤后的规则统计结果. 首先, 规则被分为三类: "BP", 表示双语短语 (Bilingual phrase) 规则; "TR", 代表 Lanchier 用到的树到树转换规则 ($\alpha = 1$);"TSR", 代表 Lanchier 用到的树序列到树序列转换规则 ($\alpha > 1$). 其次, 对每类规则按照词汇化程度又各分为三个小类: "L", 代表全词汇化 (Full lexicalized) 规则, 即叶子节点全部为终结符的规则; "P", 代表部分词汇化 (Partial lexicalized) 规则, 也即叶子节点中既包含终结符又包含非终结符的规则; "U", 代表非词汇化 (Unlexicalized) 规则, 即叶子节点全部为非终结符的规则.

表 1 整体模型中包含的规则集的统计结果

Table 1 The statistics of the rules extracted from training data

规则	L	P	U	总数
BP	4 095 127	0	0	4 095 127
TR	1 179 805	3 092 391	46 029	4 318 225
TSR	3 262 011	3 858 641	2 094	7 122 746
总数	8 536 943	6 951 032	48 123	15 536 098

表 2 测试时所用规则统计分析

Table 2 The statistics of the rules used in testing

规则	L	P	U	总数
BP	322 965	0	0	322 965
TR	443 010	144 459	24 871	612 340
TSR	225 570	103 932	714	330 216
总数	991 545	248 391	25 585	1 265 521

4.3 实验结果及讨论

4.3.1 整体性能对比

需要说明一点, Lanchier 也可以进行仅利用双语的短语对进行单调的搜索 (通过将每个词看成一棵树). 在本文实验中, 我们以下列三种规则集设置分别运行 Lanchier:

1) BP, 仅包含双语短语规则进行单调搜索;

2) TR, 仅包含树到树转换规则 (通过设置 $\alpha = 1$, 即模拟基于 STSG 的模型);

3) ALL, 全部规则.

对比结果如表 3 所示. 结果表明: 1) Lanchier-BP 由于其进行单调搜索, 性能明显低于 Moses; 2) Lanchier-TR 由于具备了 STSG 规则, 显著地超过了 Moses (相对提升 3.6 %); 3) Lanchier-ALL (STSSG 模型) 显著地超过了 Moses (相对提升 9.3 %) 以及 Lanchier-TR (相对提升 5.5 %). 这些结果经验性地验证了本文所提出的基于 STSSG 的模型的有效性.

为了进一步分析树序列到树序列转化规则对性能的贡献情况, 我们又进行了一组实验. 其中, Lanchier 的参数 α (一个树序列中包含的最大树个数) 从 1 变化到 5. 实验结果如图 5 所示. 从中我们看出, 在 α 的值逐步从 1 增大到 3 的过程中, 性能都有很明显的提升. 这些提升表明树序列规则确实对性能的提高有较大帮助. 主要原因在于 STSSG 模型相对于一般的基于句法的模型可以有效利用不满足句法限制的翻译等价对和更多的调序规则. 另外, 从图 5 还可以看出, 当 α 大于 3 时性能不再有明显变化, 这种情形和基于短语的模型中的短语长度与性能之间关系类似 (一般长度大于 3 的短语对基于短语模型性能的贡献不很明显).

[1] 2005 年度 NIST 评测相关信息可参见网址: http://www.nist.gov/speech/tests/mt/2005/doc/mt05eval_official_results_release_20050801_v3.html. 但因使用训练数据以及前后处理上的差异, 故和本文中涉及的系统无法直接比较.

表 3 在 2005 年 NIST 评测集上基于 STSSG 文法模型和两个基准系统的对比结果 (STSSG 文法模型的参数 $\alpha = 4$, 所列结果用自举重采样法获得, 具有 95 % 置信度)

Table 3 Comparison between the STSSG-based model with two baseline systems on 2005 NIST MT task ($\alpha = 4$ for Lanchier-STSSG. The results are with the 95 % confidence intervals, obtained using bootstrapping resampling.)

System	Rule set	Bleu (%)
Moses	BP	23.86 ± 0.44
Lanchier	BP	22.05 ± 0.42
Lanchier	TR	24.71 ± 0.45
Lanchier	ALL	26.07 ± 0.45

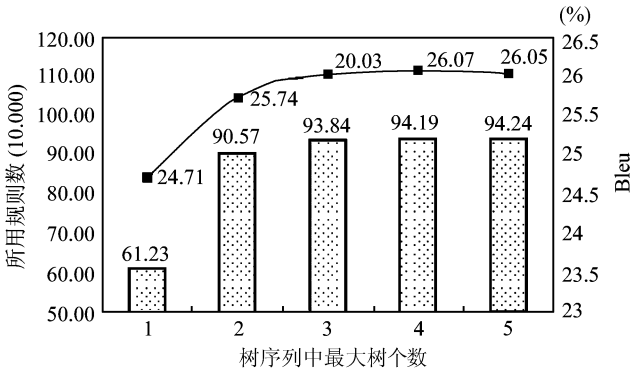

图 5 基于 STSSG 系统的性能随参数 α 变化 (从 1 到 5) 的性能变化曲线

Fig. 5 The performance curve of the STSSG-based model with α varying from 1 to 5

4.3.2 效率对比分析

在本文实验基于的服务器上 (CPU 主频 1 599 Mhz, 内存 4G), Moses 系统和 Lanchier 系统的训练时间 (包括规则抽取和翻译概率统计) (Training 用时) 以及测试集解码时间 (Decoding 用时) 对比如表 4 所示. 从表 4 中可以看出, Training 的时间上, Lanchier 耗费时间约为 Moses 的 1.3 倍; Decoding 时间上, Lanchier 耗费时间则约为 Moses 的 15.5 倍. 从此可以看出, Lanchier 较 Moses 来讲, 解码效率还显得很低.

表 4 Moses 和 Lanchier 的时间效率对比

Table 4 Time cost comparison between Moses and Lanchier

系统	Training 用时	Decoding 用时
Moses	8 小时 30 分钟	51 分钟
Lanchier	12 小时	12 小时 56 分钟

5 结论

本文提出了一种基于同步树序列替换文法的机器翻译模型. 此模型通过把树序列作为基本翻译单元, 从而可以对不满足句法限制的翻译等价对知识进行有效利用, 同时一般基于句法模型的全局调序能力和对非连续短语进行有效建模的能力也得到继承和增强. 实验结果表明本文所提出的模型的性能优于当前流行的短语模型 Moses 以及一个基于树替换文法的模型.

References

1 Wu D K. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 1997, **23**(3): 377−403

2 Chiang D. A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Ann Arbor, Michigan: Association for Computational Linguistics, 2005. 263−270

3 Liu Y, Liu Q, Lin S X. Tree-to-string alignment template for statistical machine translation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Sydney, Australia: Association for Computational Linguistics, 2006. 609−616

4 Jiang Hong-Fei, Li Sheng, Fu Guo-Hong, Zhao Tie-Jun, Zhang Min. A tree-substitution grammar based model for statistical machine translation. *Journal of Software*, 2009, **20**(5): 1241−1253
(蒋宏飞, 李生, 付国宏, 赵铁军, 张民. 一种基于同步树替换文法的统计机器翻译模型. 软件学报, 2009, **20**(5): 1241−1253)

5 Liu Y, Huang Y, Liu Q, Lin S X. Forest-to-string statistical translation rules. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic: Association for Computational Linguistics, 2007. 704−711

6 Xiong D Y, Liu Q, Lin S X. Maximum entropy based phrase reordering model for statistical machine translation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Sydney, Australia: Association for Computational Linguistics, 2006. 521−528

7 Galley M, Hopkins M, Knight K, Marcu D. What′s in a translation rule? In: Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics. Boston, USA: Association for Computational Linguistics, 2004. 273−280

8 Eisner J. Learning non-isomorphic tree mappings for machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. Sapporo, Japan: Association for Computational Linguistics, 2003. 205−208

9 Zhang M, Jiang H F, Aw A, Li H Z, Tan C L, Li S. A tree sequence alignment-based tree-to-tree translation model. In: Proceedings of the 46th Annual Meeting on Association for Computational Linguistics. Ohio, USA: Association for Computational Linguistics, 2008. 559−567

10 Fox H J. Phrasal cohension and statistical machine translation. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002. 304−311

11 Wellington B, Waxmonsky S, Melamed I D. Empirical lower bounds on the complexity of translational equivalence. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Sydney, Australia: Association for Computational Linguistics, 2006. 977−984

12 Koehn P, Och F J, Marcu D. Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics. Edmonton, Canada. 2003. 48−54

13 Bod R. Unsupervised syntax-based machine translation: the contribution of discontiguous phrases. In: Proceedings of Machine Translation Summit XI. Copenhagen, Denmark: Centre for Language Technology, 2007. 51−56

14 Li C H, Li M H, Zhang D D, Li M, Zhou M, Guan Y. A probabilistic approach to syntax-based reordering for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic: Association for Computational Linguistics, 2007. 720−727

15 Wang Bo, Yang Mu-Yun, Li Sheng, Zhao Tie-Jun. Evaluation of all-words WSD for Chinese in machine translation. *Acta Automatica Sinica*, 2008, **34**(5): 535−541
(王博, 杨沐昀, 李生, 赵铁军. 中文全词消岐在机器翻译系统中的性能评测. 自动化学报, 2008, **34**(5): 535−541)

16 Och F J, Ney H. Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania: Association for Computational Linguistics. 2002. 295−302

17 Och F J. Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. Sapporo, Japan: Association for Computational Linguistics, 2003. 160−167

18 Koehn P, Hoang H, Birch A, Callison-Burch C, Federica M, Bertoldi N. Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic: Association for Computational Linguistics, 2007. 177−180

19 Och F J, Ney H. Improved statistical alignment models. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. Hong Kong, China: Association for Computational Linguistics, 2000. 440−447

20 Zhang Y, Vogel S, Waibel A. Interpreting BLEU/NIST scores: how much improvement do we need to have a better system? In: Proceedings of the 4th International Conference on Language Resources and Evaluation. Lisbon, Portugal: Istituto di Linguistica Computazionale, 2004. 2051−2054

蒋宏飞 哈尔滨工业大学博士研究生. 主要研究方向为机器翻译和自然语言处理. 本文通信作者.
E-mail: hfjiang@mtlab.hit.edu.cn
(**JIANG Hong-Fei** Ph. D. candidate at the School of Computer Science and Technology, Harbin Institute of Technology. His research interest covers machine translation and natural language processing. Corresponding author of this paper.)

李 生 哈尔滨工业大学教授. 主要研究方向为自然语言处理和机器翻译.
E-mail: shengli@mtlab.hit.edu.cn
(**LI Sheng** Professor at the School of Computer Science and Technology, Harbin Institute of Technology. His research interest covers machine translation and natural language processing.)

张 民 新加坡信息通讯研究所高级研究员. 主要研究方向为自然语言处理和机器翻译.
E-mail: mzhang@i2r.a-star.edu.sg
(**ZHANG Min** Senior researcher at the Institute for Infocomm Research, Singapore. His research interest covers machine translation and natural language processing.)

赵铁军 哈尔滨工业大学教授. 主要研究方向为自然语言处理和机器翻译.
E-mail: tjzhao@mtlab.hit.edu.cn
(**ZHAO Tie-Jun** Professor at the School of Computer Science and Technology, Harbin Institute of Technology. His research interest covers machine translation and natural language processing.)

杨沐昀 哈尔滨工业大学副教授. 主要研究方向为自然语言处理和机器翻译.
E-mail: ymy@mtlab.hit.edu.cn
(**YANG Mu-Yun** Associate professor at the School of Computer Science and Technology, Harbin Institute of Technology. His research interest covers machine translation and natural language processing.)