

文章编号: 1003-0077(2008)02-0028-12

基于句法的统计机器翻译综述

熊德意^{1,2}, 刘群¹, 林守勋¹

(1. 中国科学院 计算技术研究所, 北京 100080; 2. 中国科学院 研究生院, 北京 100039)

摘要: 本文对基于句法的统计机器翻译进行了综述。按照模型所基于的语法不同, 将基于句法的统计机器翻译分为两大类: 基于形式化语法和基于语言学语法。对这两个不同类别, 我们分别介绍它们代表性的工作, 包括模型的构建、训练和解码器的设计等, 并对比了各个模型的优点和缺点。最后我们对基于句法的统计机器翻译进行了总结, 指出设计句法模型时要注意的问题, 并对未来的发展趋势进行了预测。

关键词: 人工智能; 机器翻译; 统计机器翻译; 基于句法的统计机器翻译; 树到串; 树到树; 依存语法

中图分类号: TP391

文献标识码: A

A Survey of Syntax-based Statistical Machine Translation

XIONG De-yi^{1,2}, LIU Qun¹, LIN Shou-xun¹

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China;

2. Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: This paper presents an overview of recent syntax-based statistical machine translation (SMT). According to the differences of grammars which translation models are based on, we classify the syntax-based SMT into two categories: formally syntax-based SMT and linguistically syntax-based SMT. For each category, we discuss the representative work, including model design, training and decoding. We also make a comparison of different models. Finally, we point out the problems on designing syntax models and give a prediction of future development of syntax-based SMT.

Key words: artificial intelligence; machine translation; statistical machine translation; syntax-based SMT; tree-string; tree-tree; dependency grammar

引言

基于句法的统计机器翻译最早可以追溯到 90 年代初, 当时同步树粘接语法 (Synchronous Tree-adjoining Grammar, 简称 STAG)^[1] 和反向转录语法 (Inversion Transduction Grammar, 简称 ITG)^[2] 相继提出来并用到机器翻译上。从时间上说, 与 IBM 提出基于单词的统计翻译模型的时间很接近, 但是基于句法的统计机器翻译研究逐渐得到人们的关注和

认可却是在 2000 年之后。即使是在这段时间, 许多研究者对基于句法的统计机器翻译仍然持观望态度, 主要是因为基于短语的统计机翻译仍然方兴未艾, 最初的基于句法的统计机器翻译系统在性能上与基于短语的统计机器翻译系统相差甚远, 再加上 2003 年由 Och 等人组织的约翰·霍普金斯大学 (JHU) 夏季研讨班“Syntax for Statistical Machine Translation”经过 6 个星期的努力之后, 发现引进诸多与句法结构相关的特征并不能显著改善翻译质量^[3]。这些使得人们对基于句法的统计机器翻译产生了怀疑^①。

收稿日期: 2007-03-12 定稿日期: 2007-09-07

基金项目: 国家自然科学基金资助项目 (60573188); 国家 863 计划资助项目 (2006AA010108)

作者简介: 熊德意 (1979—), 男, 博士, 主要研究方向为统计机器翻译; 刘群 (1966—), 男, 研究员, 博导, 主要研究方向为自然语言处理和机器翻译; 林守勋 (1948—), 男, 研究员, 博导, 主要研究方向为多媒体技术及应用系统。

① 这中间存在一个矛盾, 即理论上人们普遍认为句法对机器翻译有很大作用, 但是实际系统却没有证实这一点。这说明或者是句法对机器翻译真的不起作用, 或者是人们在利用句法信息时在方法上走了弯路。从后来的工作来看, 句法信息如何利用是一个关键问题。

但是基于短语的统计机器翻译本身存在一些固有缺陷,如短语层次上的全局重排序,短语非连续性和泛化能力问题,极大地束缚了该方法的进一步发展。这使得人们又不得不求助于句法,因为在理论上或者在人们的直觉上,引入句法结构知识有助于解决这些问题。所以纵观统计机器翻译的发展历程,可以看到,基于句法的统计机器翻译是继基于短语的统计机器翻译之后的一个新趋势。从目前的情况来看,有些基于句法的统计机器翻译系统在性能上已经明显超过了基于短语的系统,如 2005 年 NIST^[4] 机器翻译评测中的 Chiang 的 Hiero 系统,2006 年 NIST 机器翻译评测中的 ISI 的系统和中科院计算所的系统,在性能上它们接近甚至超过了最好的短语系统。

将句法知识引入到统计机器翻译系统中,存在多种不同的方法,如在单词对齐模型中引入句法知识^[5],在翻译之前利用句法知识调整源语言语序^[6],在翻译之后利用句法知识做 reranking^[3]等,本文只讨论翻译模型本质上就是基于句法的,并称之为基于句法的统计机器翻译。本文延续 Chiang^[7] 的分类思想,根据翻译模型所依赖的语法是否包含语言学知识将基于句法的统计机器翻译粗略分为以下两类:

1) 基于形式化语法的: 该类翻译模型建立在形式化语法的基础上,但并不包含人类语言学知识,如短语标记,词与词之间的依赖关系等;

2) 基于语言学语法的: 该类模型建立在语言学语法基础上,将人类语言学知识包含到模型中。根据所采用的结构树形式的不同,又可以将它分为以下两类:

a) 基于短语结构树的: 该类模型通过短语结构树,将短语的句法标记及标记之间的依赖关系等语言学知识引入到翻译过程中;

b) 基于依存树的: 该类模型通过依存树,将词与词之间的依赖约束关系等语言学知识引入到翻译过程中。

在下文中,我们将依次介绍这几类不同的基于句法的统计机器翻译。在“基于形式化语法的统计机器翻译”中,主要介绍 Chiang 的层次短语模型和吴德凯的 ITG 模型;在“基于短语结构树的统计机器翻译”中,将分别介绍树到串,串到树,和树到树的模型;在“基于依存树的统计机器翻译”中,主要介绍 Lin, Quirk 和 Ding 等人的工作。最后给出总结。

1 基于形式化语法的统计机器翻译

基于形式化语法的统计机器翻译模型,只用到某种形式化的语法体系,但是该语法中并不包含任何语言学知识,如一些语言学标记和关系等。当然,这些模型本身并不限制利用语言学知识,但是模型设计者往往考虑到模型复杂度,将模型退化到不包含任何语言学知识的形式化语法上。与基于单词/短语的模型相比,该类模型最明显的特点是借用了形式化语法的结构,从而使得翻译过程是层次化的,是有结构的。虽然这种结构并不体现真正的语言学结构,但是优点仍然是非常明显的:

1) 首先,层次化结构使得处理复杂的远距离重排序变得更为可行,因为在层次化结构的更高层次上,一些单词层面上的远距离重排序变成了局部排序。

2) 其次,层次化结构自然而然地要引入非终结符,即变量,从而使得模型能够处理非连续短语,同时具有一定的泛化能力。

可以说,基于形式化语法的统计机器翻译已经具备了基于句法的统计机器翻译的一般优点,只是这些优点没有同语言学知识相关联。

本节要介绍的两个基于形式化语法的模型——吴德凯的 ITG 模型和 Chiang 的层次短语模型——本质上都可以看作是同步上下文无关文法(Synchronous Context-Free Grammar,简称 SCFG),因此有必要先简单介绍一下 SCFG,然后在此基础上展开对这两种模型的探讨,最后简单介绍基于 SCFG 的解码器设计。

1.1 同步上下文无关文法(SCFG)

同步上下文无关文法(SCFG)^[8]是对单语上下文无关文法(CFG)的双语扩展,比如对于单语的 CFG 规则:

$$VP \rightarrow V NP$$

扩展为同步 CFG 的双语规则可以是:

$$(VP \rightarrow V_1 NP_2, VP \rightarrow NP_2 V_1)$$

其中数字标号代表一一对应关系,具有相同标号的非终结符相互对应。简单而言,同步上下文无关文法可看作是一个三元组:

$$SCFG = (G_1, G_2, \infty)$$

其中 G_1, G_2 是单语的上下文无关文法, ∞ 是 G_1, G_2 的对应关系。可以看出,SCFG 和 CFG 一

样,描述了语言的递归语法关系,不同的是,它同时描述了两种语言的语法结构,并且给出了这两种语言结构上的对应关系。

如果按照形式语法的定义,则 SCFG 是这样一个四元组:

$$SCFG = (N, T, P, S)$$

其中 N, T 分别是非终结符和终结符的有限集合, S 是特定的开始符, P 是重写规则的有限集合, 重写规则具有下面的形式:

$$(A \rightarrow \alpha, B \rightarrow \beta, \infty)$$

其中 A, B 是非终结符, α, β 是非终结符和终结符组成的字符串, ∞ 是 α, β 中非终结符的一一对应关系。 A, B 可以相同, 相同时可以将重写规则简写为

$$X \rightarrow (\alpha, \beta, \infty)$$

那么怎样用 SCFG 进行推导呢? 和 CFG 一样, 它也是从开始符开始, 然后不断用重写规则重写产生式中的非终结符, 注意每次展开的非终结符必须是对应在一起的, 不能同时展开没有对应关系的两个非终结符, 如:

$$(S_1, S_2) \rightarrow (NP_1 V_2 NP_3, NP_1 NP_3 V_2)$$

如果采用最左推导, 那么只能同时展开两个 NP_1 , 而不能一边展开 NP_1 , 而另一边展开 NP_3 。

1.2 吴德凯的 ITG 模型

ITG 是最早将同步语法用于统计机器翻译的模型之一^[2,9]。它可以看作是 SCFG 的一个约束简化版本, 其中一个最重要的假设是, 同步规则右边两种语言的语序只存在两种可能性: 保序或者逆序。保序是指源语言和目标语言的语序完全一致; 逆序则是指它们的语序恰恰相反。基于该假设, SCFG 重写规则中的对应关系 ∞ 可以简化为两种, 分别用 $[]$ 和 $\langle \rangle$ 来表示, 前者表示保序, 后者表示逆序。虽然这种重排序假设很严格, 但是该模型仍然覆盖了语言间的大部分语序差异^[2,10]。

为了降低计算复杂度, ITG 可以进一步退化成为 BTG (Bracketing Transduction Grammar), BTG 只有一个非终结符, 其规则也只有下面 3 种:

$$A \xrightarrow{[]} (A^1, A^2) \quad (1)$$

$$A \xrightarrow{\langle \rangle} (A^1, A^2) \quad (2)$$

$$A \longrightarrow (x, y) \quad (3)$$

规则(1)用于保序地合并两个相邻成分, 规则(2)用于逆序地合并两个相邻成分, 规则(3)用于翻

译源语言的单词/短语。

ITG 除了作为一种同步 CFG 语法之外, 它还可以作为一种重排序约束条件^[11,12] (Reordering Constraint)。基于 ITG, 对相邻的串进行重排序时只需要考虑两种情况, 保序或者逆序(即交换它们的顺序)。这种策略非常简单, 但是这种两者皆有可能的“模棱两可”也成为该策略一个主要缺陷。到底是保序还是逆序, ITG 并没有明确规定, 最终往往要借助于 N-gram 语言模型来确定, 或者事先根据不同语言之间语序的差异情况设置一个保序或逆序的先验值, 比如汉语—英语语序比较相似, 保序的先验值可以设为 0.8, 逆序的先验值可以设为 0.2。这两种做法都存在问题。首先语言模型只能考虑到相邻的几个边界单词, 对较长的串, 语言模型并没有能力正确区分保序和逆序, 因为这两种语序调整之后的语言模型的值可能并没有太大的差别。而设置先验的保序或逆序概率, 其准确性就更低了。

1.3 Chiang 的层次短语模型

层次短语模型^[7,13]在重排序约束方面比 ITG 模型要弱一些, 允许的排序可能性更多一些。其规则形式为:

$$X \rightarrow (\alpha, \beta, \infty)$$

其中 X 为非终结符, α, β 是非终结符和终结符组成的字符串, ∞ 是 α, β 中非终结符的一一对应关系。下面给出几个规则的例子:

$$X \rightarrow (yu X_1 you X_2, have X_2 with X_1)$$

$$X \rightarrow (X_1 zhiyi, one of X_1)$$

为了增强模型的鲁棒性, 以及与短语模型的做法保持一致(将句子切分为短语, 然后翻译各个短语), Chiang 在他的层次短语模型中引入了以下两条规则:

$$S \rightarrow (S_1 X_2, S_1 X_2)$$

$$S \rightarrow (X_1, X_1)$$

这两条规则称为“glue”规则, 其作用是允许模型能对源语言进行短语切分然后顺序合并它们的译文。显然这两条规则类似于 ITG 中的保序规则, 如果只允许这两条规则, 再加上短语模型中的短语对, Chiang 的层次短语模型就退化为单调解码的短语模型了。

由于该模型是形式化的句法模型, 因此不需要对源语言或目标语言做任何句法分析, 就可以从平行语料中抽取这些规则。抽取的方法是:

1) 首先从单词对齐的双语语料中抽取短语

(f_i^s, e_i^s) , 并将这些短语改造为完全词汇化的基本规则 $X \rightarrow (f_i^s, e_i^s)$; 这里短语的定义和抽取方法与短语模型中的定义和方法是一致的。

2) 然后从已抽取的短语中构建带变量的规则: 如果一个短语中包含有其他短语, 则将该短语里嵌套的子短语替换为变量 X 。

按这种方法获得的同步规则数量非常大, 为了降低复杂度, 需要定义一些条件来约束同步规则的抽取。文献[7]采用的主要约束条件有:

1) 规则的长度在源语言端不能超过 L (完全词汇化的基本规则不能超过 L_1 , 带变量的规则不能超过 L_2);

2) 规则最多只能含有 N 个非终结符, 即变量 X 的数目不能超过 N ;

3) 规则源语言端中不允许存在两个相邻的非终结符;

4) 规则必须含有一对对齐的单词。

那么如何在抽取出来的同步语法上构建概率化的模型呢? 借鉴基于短语的统计机器翻译中经常使用的对数线性(Log-Linear)模型^[14], 对于从双语语料中抽取出来的每条规则, 其分数可以按下式计算:

$$\text{score}(X \rightarrow (\alpha, \beta)) = \prod_i f_i(X \rightarrow (\alpha, \beta))^{\lambda_i}$$

其中 f_i 为定义在规则之上的特征, λ_i 为相应特征的权重。文献[7]定义了如下特征:

1) 两个方向上的短语翻译概率 $p(\alpha|\beta), p(\beta|\alpha)$;

2) 按照文献[21]中的方法计算的 IBM 模型 1 的概率, 即两个方向上的词汇化概率 $p_{\text{lex}}(\alpha|\beta), p_{\text{lex}}(\beta|\alpha)$;

3) 短语个数;

而对于“glue”规则, 其分数定义为 $\exp(-\lambda_g)$, 这可以看作是对保序合并两个相邻变量的一种惩罚。

1.4 基于同步上下文无关语法的解码器设计

基于 SCFG 的解码过程实际是一种扩展的短语分析过程。它可以分为两个基本步骤: 1) 源语言句法分析; 2) 将源语言句法分析树转换为目标语言树, 并从中导出目标译文。之所以可以这样简单的分解, 是因为在 SCFG 框架下, 经同步规则推导得到的源语言树和目标语言树是同构的, 这是 SCFG 的一个重要假设, 也是它的一个重要缺陷, 因为两种语言在语言学意义上的句法分析树往往是不同构的。

给定一个源语言句子, 我们可以利用同步规则中源语言端的语法, 基于句法分析的常见算法, 如

CKY, 构建源语言端的分析树。这个过程中, 我们可以对同步规则进行二分化^[15], 将同步规则中源语言端和目标语言端的语法转化成乔姆斯基范式。但要注意的是, 并不是所有的同步规则都可以被二分化。

得到源语言端的树之后, 利用同步规则, 将其转换为目标语言树。转化过程中根据同步规则中的对应关系 \propto 对目标语言做相应的语序调整。值得注意的是, 同一棵源语言树可以转换成多棵不同的目标语言树, 因为很多同步规则具有相同的源语言端, 不同的目标语言端。

在实际的解码器^[2,7]中, 这两个步骤往往是同时完成的, 即解码过程同时得到源语言树和目标语言树。并且为了选择最优的推导序列, 解码过程还利用目标语言端的语言模型对中间的树结构进行选择。这在一定程度上增加了解码器的计算复杂度。

2 基于语言学语法的统计机器翻译

基于语言学语法的统计机器翻译, 不仅采用了形式化的句法体系, 同时本身也包含了丰富的语言学知识。语言学知识既包括语言内本身的结构知识, 也包括源/目标语言之间的结构和翻译等价物之间的映射关系。我们都知道, 语言结构可以用不同的形式来描述, 在自然语言处理中最常见的两种形式是依存树和短语结构树。依存树描述了词与词之间关系, 短语结构树则描述了句子的组成成分及各成分之间的关系。相比较而言, 依存树更体现了句子的内部语义结构, 而短语结构树则更多地体现了句子内部的句法结构。由于依存树和短语结构树存在这种天然的不同性, 本章分开介绍与它们相关的模型。2.1 节着重介绍基于短语结构树的统计翻译模型, 2.2 节介绍基于依存树的统计翻译模型。

2.1 基于短语结构树的统计机器翻译

基于短语结构树的统计翻译模型, 可以大致分为两大类, 一类是树一串模型, 另一类是树一树模型。前者在短语结构树和串之间寻找映射关系, 后者在短语结构树之间寻找映射关系。由于两种语言的不同构性(即语言的结构存在差异)导致了树与树之间映射的复杂性, 目前还没有看到基于树一树模型的成熟的统计翻译系统, 树一树模型更多的是处在理论探索阶段。但这并不妨碍我们介绍其中一些理论上比较成熟、并被看好的树一树模型。

树—串模型,相对于树—树模型而言,一方面复杂度要低,另一方面只需要某一端的语言有句法分析器,这对于那些缺乏句法分析资源的语言而言是非常合适的。同时,目前那些基于短语结构语法的、成熟的、性能优越的统计翻译系统大多是基于树—串模型的。所以本节会较详细地介绍与树—串模型相关的工作。

下面首先探讨树—串模型,并按照模型是基于源语言结构树还是基于目标语言结构树,将树—串模型分为树到串的模型和串到树的模型。注意,这里的方向是按照源语言—目标语言的顺序来说的,并不是基于噪声通道的方向。在此基础上,介绍一些有代表性的工作。最后介绍几种树—树模型。

2.1.1 串到树模型

该类模型中最典型的代表模型是南加州大学信息科学研究所(ISI/USC)提出的树—串模型。它们的基本思想是:目标语言端是有短语结构树的,按照噪声信道模型来解释的话,就是目标语言的树经过有噪声的通道后被异化成源语言的串,解码的任务就是将源语言的串还原成目标语言的树。最初,Yamada等提出的模型,试图通过一些操作去捕捉噪声通道中的异化过程,并对这些操作赋以不同概率;后来 Galley 等人将这个异化过程用表达能力更强的规则来描述,规则的一端是带变量的源语言串,另一端是目标语言相应的子树结构。Marcu 等人又在 Galley 的工作基础上,加入了更多的特征,并且使得模型可以更好地融合基于短语的统计机器翻译模型中的短语。经过逐步的发展,ISI 提出的树—串模型不断得到完善和改进,以至于他们的系统在 2006 年 NIST 评测中在汉—英翻译中名列第一。下面将按照上面所述的 3 个不同阶段来介绍 ISI 的树—串模型。

• Yamada 等人的工作^[16,17]

Yamada 等最初提出的模型可以看作是 ISI 树—串模型的一个开端,该模型首先对目标语言进行句法分析,得到目标语言的结构树;结构树通过噪声通道时,每个节点都要经过一系列所谓的“通道操作”,使得最终将所有叶子节点串在一起能得到源语言的串。这些通道操作包括:

1) 插入操作:选择一个源语言单词,将它插到该节点的左边,或右边,或者干脆不做任何插入。

2) 重排序操作:对该节点的所有孩子节点(包括被插入单词形成的节点)进行重排序,如果有 N 个孩子节点,则要考虑所有 $N!$ 种可能的重排序。该操作只对非终结符有效。

3) 翻译操作:将目标语言结构树的叶子节点翻译为相应的源语言单词。

每个操作都有自己的概率,模型通过 EM 算法估计这些操作的概率。解码的过程,就是从源语言的句子反向搜索出目标语言的结构树,算法上可以通过 CYK 来实现。

这些操作可以通过同步 CFG 规则来模拟,如在节点 σ 的左边插入单词 w ,同步 CFG 规则可以表示为:

$$\sigma \rightarrow (w\sigma_1, \sigma_1)$$

因此可以说,Yamada 的模型等价于某种 SCFG 模型。但是我们知道 SCFG 同 CFG 一样,是有一些缺点的,如它能捕捉的上下文是有限的,由此导致的是 SCFG 只能描述单层树结构,而不是多层。这使得重排序局限于同一个父节点的不同孩子节点之间,大大地限制了模型的表达能力。另外 Yamada 的模型是建立在单词的基础上,这也限制了模型的性能。虽然 Yamada 在这两方面都做了一些改进,如引入扁平化操作使得多层之间的重排序也是可以做到的,允许叶子节点是短语等,但是这些改进终究还是在 SCFG 的框架下,因此一些缺陷还是不能从根本上解决。

继 Yamada 之后,Galley 和 Marcu 等人对串到树的模型做了一些突破性的工作。Galley 一个主要工作是引入了能够描述多层树结构的规则,大大扩展了模型的表达能力。Marcu 的主要工作是对 Galley 的模型继续改进,引入更多特征,并使其能与短语兼容。

• Galley 等人的工作^[18,19]

该工作的一个基本思想是把模型建立在能够描述多层树结构的转换规则上。为此,首先对目标语言进行句法分析,得到目标语言的结构树;然后将目标语言的结构树和源语言句子对齐,即目标语言叶子节点对应到源语言的单词上,这个只需要将现有的单词对齐投射到树上就行了;然后从这种串到树的对齐中,按照一定的算法,自动抽取一些规则。这些规则表示了短语结构子树和串之间的对应关系,它们可以简单分为三类:

1) 翻译单词或短语的简单规则,如 NP-C(NPB(DT(this) NN(address)))→这个地址;

2) 源语言端带有非终结符的规则,如 NP-C(NPB(PRP(my) x0;NN))→我的 x0;

3) 合并规则,即源语言端全部由非终结符组成,如 VP(x0; VBZ x1; NP-C)→x0 x1。

文献[18]中抽取的是最小规则,即和该源语言

片段、目标语言结构树以及它们之间的对齐保持一致的最小的规则,它们不能再被拆分为其他规则。而在文献[19]中,他们不仅抽取了最小规则,而且也抽取了由最小规则组成的复合规则,并且发现这些复合规则对系统的性能提升有很大的帮助。从某种程度上说,可以把最小规则类比为单词,而把复合规则类比为短语,我们都知道基于短语的 SMT 要明显优于基于单词的 SMT,这是因为短语捆绑了更多的上下文信息,如局部的单词选择和短语内部的单词顺序等,复合规则相对于最小规则而言,也包含了更多的上下文信息,这是复合规则提升性能的主要原因。除此之外,文献[19]还详细说明了如何估计规则的概率以及 EM 训练过程。

解码的过程类似于单语分析的过程,即对源语言端进行“句法分析”,用转换规则右边去匹配源语言,用左边去生成目标语言的结构树。

• Marcu 等人的工作^[20]

Marcu 等人^[20]提出了一个和文献[19]工作非常类似的串到树的模型: SPMT 模型。虽然该模型表达能力不及文献[19]的模型强^①,但 Marcu 等人在 ISI 串到树系列模型上引入了一些新工作,主要有: 1) 使模型能够兼容非句法短语; 2) 采用更多的特征函数估计规则的概率。

由于在抽取规则时做出了如下限制: 如果一个节点包含在某条规则中,那么该节点所有的姐妹节点也包含在该规则中。这就表明了抽取算法不能抽取那些没有和目标语言结构树对齐的短语,这些短语由于不符合句法,所以称之为非句法短语。非句法短语或者跨越两个不同的句法短语,但是又没有完全覆盖这两个句法短语(比如“布什总统 发表”,“布什总统”是一个完整的名词短语,“发表”则只是动词短语的一部分),或者是一个句法短语的部分孩子节点组成的短语。在基于短语的统计机器翻译系统中,非句法短语得到了大量的应用,事实上证明这些短语对系统性能有很大的影响^[21]。所以在 syntax-based 系统中融合 phrase-based 系统中的短语是非常重要的。文献[20]中一项工作就是使他们的系统能够兼容非句法短语,他们的做法是: 对非句法短语,创建一个伪的、非句法的非终结符来覆盖它,由此构建一条转换规则;同时创建另一条相配对的规则,该规则描述了非句法的非终结符如何与其他真正的非终结符组合成句法树。

另外一个工作是估计规则的概率。除了 syntax-based 特征之外,文献[20]借用了大量的 phrase-based 模型中特征函数,如基于 IBM 模型 1

的词汇化概率、语言模型、单词惩罚等。

2.1.2 树到串模型

下面要介绍的树—串模型,与 ISI 模型相比,相同的是它们都是用概率化的规则来描述树串之间的转换关系的;但不同的是,它不是从源语言的串到目标语言的结构树,而是相反,从源语言的结构树到目标语言的串。树到串模型典型的工作可参阅文献[22~24]。由于它们的相似性,这里主要介绍文献[23]中提到的树模板模型。

• 刘洋等人的树模板模型^[23]

该方法的基本思想: 首先用句法分析器获得源语言的短语结构树,然后利用树到串对齐模板(TAT)将源语言的树映射到目标语言上,因此解码过程实际上更像一个树到串的转换过程。下面将首先给出树到串对齐模板的定义并说明如何通过 TAT 实现将源语言的短语结构树转换为目标语言的串,然后阐述 TAT 的抽取和解码器的设计。

树到串对齐模板是一个三元组 $\langle T, S, A \rangle$, T 为源语言的句法结构子树, S 为和 T 的叶子节点相对应的目标串,该串可以含有终结符(目标语言单词),也可以含有非终结符(对应于 T 中的非终结符叶子节点), A 为 T 的叶子节点和 S 中元素(终结符和非终结符)的对齐关系。

图 1 给出了几种不同的 TAT。

任何一个 TAT 必须满足以下两个条件:

- 1) T 必须是源语言结构树中的一棵子树;
- 2) 单词对齐的约束,即 T 中的任何一个叶子节点对应的元素必须在目标串 S 内,而目标串 S 内的任何一个元素(单词或非终结符变量)也必须对应到 T 中的一个叶子节点上。

按照这两个条件,从语料库中抽出来的 TAT 的数目仍然是非常庞大的,为此,继续增加以下约束条件:

- 1) S 中的第一个和最后一个元素必须对应到 T 中的叶子节点上,而不能对空。
- 2) T 的高度不能超过 h 。
- 3) T 中任何一个内部节点的直接孩子数不超过 c 。

由于 TAT 限制 T 必须为子树,因此这种抽取方法会导致一些非结构成分的短语抽取不出来。所

^① 两个模型在目标语言端都能表示多层树结构,但是在源语言端,SPMT 模型要求短语必须是连续的,而文献[19]中的模型可以允许非连续短语存在。

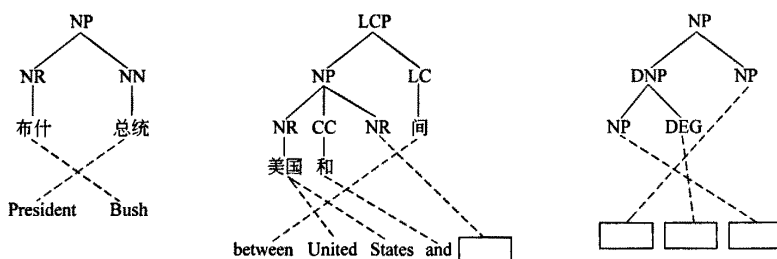


图1 不同的 TAT 样例

有 TAT 抽出来后,就可以估计相应的概率,如模板的翻译概率,词汇化概率等。

解码时,自底向上遍历源语言结构树中的每个节点,对以当前节点为根节点的子树,解码器搜索所有匹配的 TAT,通过这些 TAT 得到该节点相应的候选译文,并将它们存贮到该节点对应的堆栈里。更高层节点的译文由 TAT 确定的低层孩子节点对应的译文组装得到。

为了减少搜索空间,提高翻译速度,文献[23]做了以下几点限制:

1) 限制每个节点可匹配的 TAT 数目;

2) 限制每个节点对应的堆栈中的候选译文的数目,保留最好的前 M 个。

在自底向上的搜索过程中,解码器采用动态规划思想,保存每个部分候选译文最左/右端的 N 个单词,N 取决于语言模型的阶数,对于最左/右端的 N 个单词相同的译文,解码器合并它们,只保留概率最高的一个。

2.1.3 串到树 vs 树到串

上面分别介绍了串到树模型和树到串模型,它们最本质的不同点在于是利用源语言的结构树还是利用目标语言的结构树。在这里,存在一个争论的问题,即到底哪一个更好些。目前还没有实验结果给出一个令人信服的答案,我们只能说这两种模型都有各自的优缺点:

1) 串到树模型充分利用了目标语言的句法结构信息,使得生成的目标语言更符合语法,但是它完全忽略了源语言的结构信息对翻译所起的作用。

2) 树到串模型则充分利用了源语言结构信息,这些结构信息可以引导翻译过程,同时对重排序非常有帮助,但是它并不能保证生成的目标语言合乎语法。

可以看到,两者各自的优点恰恰是对方的弱点,也许只有树一树模型可以将两者的优点统一起来。

但是树一树模型需要两端都有较好的句法分析资源,而实际情况中,许多语言对可能达不到此要求。所以,如果在源语言端有很好的句法分析器可利用,那么我们可以建立树到串模型,否则建立串到树模型。

2.1.4 树一树模型

要在两种语言之间建立树一树模型,首先要克服的一个问题是语言间的结构性差异问题^[25]。因此衡量一个树一树模型的好坏,首先要看它在多大程度上允许这种结构性差异的存在,或者说它描述结构差异的能力有多强。之前提到的同步上下文无关文法(SCFG),由于它只能描述单层的树结构,因此能够捕捉到的结构差异也限于单层结构内,即同一个父节点下孩子节点间的差异,如需要插入一个新的孩子节点,或者需要在孩子节点间调整语序。这种表达结构差异的能力还不是很强,因为语言间的差异往往涉及到多层树结构之间的变化。

基于树一树模型的统计机器翻译解码时,通常有两种做法。一种是首先通过句法分析器得到源语言的树,然后通过树到树的映射规则或者树到树的转录机^[26]将源语言的树转化成目标语言的树。另一种是一开始就没有源语言的树,在给定源语言句子的基础上需要做同步分析,同时得到源语言和目标语言的树。这两种做法实际上是相通的,它们都需要一部概率化的同步语法,实现树到树的配对映射。不同的是,是否能够获取到外部句法分析器的支持。对于第一种方法,解码的过程就是树转换的过程,对于第二种方法,解码的过程则是分析的过程^①。

由于目前基于树一树模型的实际系统还不是很

① 这里我们不能说该过程就是同步分析,因为同步分析指的是在同时给定源语言和目标语言的情况下,分析得到一棵树,覆盖两边的句子。这里的分析实际上还是一种单语分析,只不过分析结果是目标语言的句法树。

多,大部分工作还是理论上的探索,所以这里并不对所有相关的工作全面展开描述,只选择其中一些和树相关的同步语法进行概要的探讨。目前和树相关的并应用于机器翻译的同步语法理论包括同步树粘接语法(STAG)、同步树替换语法(STSG),多文本语法(MTG)等。除此之外,还有一些其他相关工作,如文献[27]引入克隆(Clone)操作,以解决树的非同构问题,文献[26]对树到树的转录机做了详细的描述等。在下面,本文只介绍同步树粘接语法(STAG)和同步树替换语法(STSG)。

• 同步树粘接语法(STAG)^[1]

树粘接语法(TAG)^[28]是由一些基本树构成的,包括初始树和辅助树,它通过定义在基本树上的两种操作——替换和粘接——完成树结构的推导。相对于CFG而言,在描述自然语言结构方面,TAG有许多优点,如有较大的局部域(Domain of Locality),也就是说可以包含较大的上下文语境,通过替换操作可以很好地表示句子的依存结构等。

同步树粘接语法(Synchronous TAG)的基本元素为分别定义在源语言端和目标语言端的基本树组成的树对,树对中的每个对应的非终结符节点之间都有一个链接(这种链接可以是一对一的链接,也可以扩展到多对多的链接)。STAG中操作也来自于TAG中的两个操作:替换和粘接;STAG的同步表现在这些操作(替换或者粘接)必须同步作用在树对中的任何一对有链接的节点上。STAG的推导过程如下(假设开始的树对为(A,B)):

- 1) 任意选择树对中的一个链接,假设为节点对 (n_1, n_2) ;
- 2) 从STAG中选择一个与 (n_1, n_2) 相匹配的基本树对 (b_1, b_2) ;
- 3) 同步执行相关的操作(替换或者粘接),将 b_1 和 b_2 分别作用在 n_1 和 n_2 上,得到新的树对 $(b_1(A, n_1), b_2(B, n_2))$ 。

以上的过程不断重复,直至不再有操作可以发生。

• 同步树替换语法(STSG)^[29]

Eisner在2003年提出了一种同步树替换语法(Synchronous Tree Substitution Grammar,简称STSG)的构想,试图解决机器翻译中树的非同构映射问题。在上几节的讨论中,我们知道同步上下文无关文法由于缺乏较强的结构差异表达能力,以至于它生成的树在结构上必须是同构的(如

果允许空节点存在,那么源语言和目标语言两边的树的节点数可以不一样),这种同构性限制使它不能够完全描述真实的语言对。树替换语法和树粘接语法比较类似,都有较大的局部域,因此基于树替换语法的同步语法可以很好表达树的非同构性。

TSG比TAG少了一个基本操作:粘接,树的推导过程只依赖于替换操作。而同步TSG每次都选择一个相连接的前端非终结符,将配对的基本树替换到该非终结符对下面,这个过程不断重复,直到生成完整的树对。Eisner在文献[29]中给出了TSG的分析算法,STSG的EM训练算法和基于STSG的解码算法。核心思想与CYK算法类似,这里不再赘述。值得一提的是,TSG语法也可以用在依存树结构上,因此可以在TSG上构建基于依存语法的统计机器翻译(见下节)。

2.2 基于依存树的统计机器翻译

在介绍基于依存树的统计机器翻译之前,有必要先简要介绍一下依存树。依存树中的每个节点对应于句子中的一个单词,这和短语结构树是不同的,短语结构树中只有叶节点和单词对应。依存树中每条有向边代表一对单词之间的关系,方向从中心节点指向修饰节点,除了根节点之外,每个节点有且只有一条有向边指向它。图2是同一个句子对应的短语结构树和依存树。

近年来,依存语法逐渐得到一些研究者的青睐。和短语结构语法相比较,依存语法存在一些适合于机器翻译的特性:

- 1) 首先依存语法天生就是词汇化的。词汇化不仅在句法分析中发挥了很大的作用,在统计机器翻译中也发挥了很大作用,如在基于短语的统计机器翻译中,引入短语的词汇化概率可以极大地提升系统性能,基于词汇化的重排序模型也大大优于基于距离的重排序模型。

- 2) 其次依存语法体现的更是一种语义上的关系,这种语义约束往往能更直接快速地作用到相关的成分上,如动词在短语结构树中,往往距离宾语较近,而距离主语较远,但是在依存树中,它和两者具有相等的距离。语义上的约束,有时候更有利于选词和调整语序,因为有很大一部分译文选择和语序关系直接和语义相关。基于依存语法的统计机器翻译很容易将语义知识引入到翻译中来^[30]。

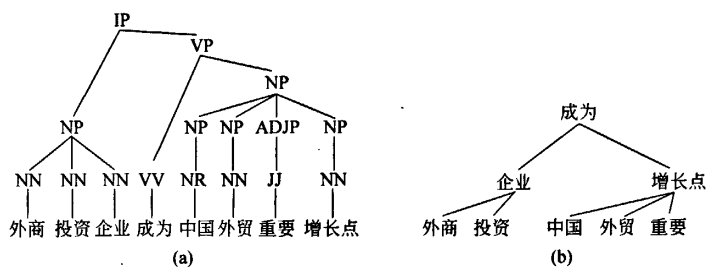


图2 短语结构树(a)和依存树(b)

3) 最后,文献[31]中指出,和短语结构树相比,依存树能更好地减小不同语言之间树结构的差异性。也就是说,同一个互为翻译的句对,它们短语结构树的差异比依存树的差异要大。依存树的这种特点,使得基于依存树的统计机器翻译在解码过程遇到的结构调整问题要少。

类似于基于短语结构树的统计模型,基于依存树的统计模型也可以分为树-树模型和树-串模型。但是目前基于依存树的工作基本上都可以看作是树-树模型,所以这里不再细分。本节对 Lin 提出的基于路径的转换模型,Quirk 等人提出的基于依存稚树(Treelet)统计句法翻译模型,以及 Ding 等人提出的基于同步依存插入语法的翻译模型进行逐一介绍。

• Lin 的基于路径的转换模型^[32]

Lin 在文献[32]将路径定义为依存树一系列的节点和边组成的序列 $n_1, \dots, n_k, \dots, n_m$ 。其中对任意的 $i \in [1, k)$,都存在一条有向边 $\langle n_{i+1}, n_i \rangle$, n_{i+1} 为 n_i 的中心节点;而对任意的 $i \in [k, m)$,都存在一条有向边 $\langle n_i, n_{i+1} \rangle$, n_i 为 n_{i+1} 的中心节点。即在节点 n_k 之前每个节点,紧跟其后的一个节点是它的中心节点,而 n_k 之后(包括 n_k)的每个节点,都是紧随其后的节点的中心节点。对每条路径,创建一个转换规则。转换规则是一个二元组,一边是源语言的路径,另一边是该路径对应的目标语言依存树片段,目标语言单词间的依存关系根据它们对应的源语言单词之间的关系来确定。转换规则可以从单词对齐的平行语料中抽取。为了提高规则的泛化能力,可以将路径中的一个末尾节点用通配符或其对应的词性标记代替。

翻译的过程如下:

- 1) 分析源语言句子,得到其依存树;
- 2) 从源语言依存树中抽取所有可能的路径,并搜索与这些路径相匹配的转换规则;
- 3) 找到一个转换规则序列,使得:

a) 它们覆盖了整棵源语言依存树;

b) 转换规则中的目标语言依存树片段可以组装成一棵目标语言依存树;

c) 从上述目标语言依存树中,挑选概率最大的一棵树作为最终的目标树;

4) 从目标树中导出译文。

从总体上说,基于路径的转换模型还是比较简单的,它没有使用目标语言的语言模型,这对最终的性能会有很大的影响。另外,它所使用的重排序策略也是基于启发式的,基本上可以看作是一种保序调整语序的策略。

• Quirk 等人的基于依存 treelet 的统计句法翻译模型^[33]

Quirk 等人的工作可以看作是对 Lin 的工作的延伸和扩展。首先,源语言这边不再是路径,而是比路径更复杂的稚树(Treelet)。稚树是依存树中任意联通的子图,这种定义要比基于短语结构树的树-串模型中的子树要更灵活,限制条件更少,覆盖范围更广。因为子树的定义中要求:如果一个节点包含在子树中,那么该节点的所有姐妹节点都包含在该子树中。但是稚树没有这样的限制条件,也就是说任何节点可以连同它的部分姐妹节点被包含在同一棵稚树中。稚树的优点是它可以将短语系统中很多短语(并没有和句法意义上的短语边界对齐)自然地包含到基于句法的统计翻译系统中。

其次,和 Lin 的工作相似,文献[33]也只需要对源语言端进行依存句法分析,但是和 Lin 不同的是,目标语言的依存关系不是在抽取转换规则时得到的,而是在这之前得到的。得到源语言依存结构之后,通过单词对齐关系,将依存结构映射到目标语言上得到目标语言的依存树,这样做的好处可以通过定义一些启发式规则对目标语言的结构进行调整。

第三,文献[33]构建了一个统计的重排序模型。在解码过程中,当使用一个匹配上的稚树对时,在目标语言树中,需要确定那些没有被该稚树覆盖的节

点的顺序,这个顺序主要是相对于中心节点而言的。Quirk 等人构建了一个统计模型,计算各种相对顺序的概率。在计算概率时,该模型利用了多种上下文信息,如中心节点和修饰节点的词汇项,或者它们的词性标记等。

最后,文献[33]使用了对数线性模型来构建自己的模型,将语言模型、稚树对互译概率模型、重排序模型等子模型加权成一个更大的模型,各子模型的参数可以通过最小错误率^[34]来训练。

解码过程类似于 CYK 算法,按自底向上的顺序遍历源语言依存树中的每个节点,查找以当前节点为根节点、匹配的稚树对,对没有被稚树对覆盖的节点,根据重排序模型调整其相对顺序。当源语言依存树的根节点翻译完之后,目标语言的依存树也构建完毕,从中可以直接获得目标译文。

该模型的缺点是:1)稚树的规模非常庞大;2)解码时,由于要考虑没有被覆盖节点的各种可能顺序,搜索空间非常大。

• Ding 等人的基于同步依存插入语法的统计翻译模型^[35]

Ding 等人提出了基于同步依存插入语法的统计翻译框架。同步插入语法和同步 TAG 比较类似,同样也有基本树的概念和两个基本的操作:替换和粘接,只不过这里的同步插入语法是定义在依存语法上的,所以需要额外地保存单词在依存树中的相对位置。模型需要在两边都有依存树的平行语料库中抽取同步依存插入语法,可以通过 EM 算法实现语法的自动获取和概率化。整个模型是一个树-树模型,解码时通过同步依存插入语法,将源语言的依存树转换为目标语言的依存树。

3 总结

本文对基于句法的统计机器翻译进行了系统的介绍。按照各种模型是否采用了语言学知识,将基于句法的统计机器翻译分为基于形式化语法的和基于语言学语法的。进一步,将基于语言学语法的统计机器翻译按照所依赖的结构树的不同又细分为基于短语结构树的和基于依存树的。

可以看到,基于句法的统计机器翻译的本质就是在翻译过程中引入句法结构,可以在源语言端引入,也可以在目标语言端引入,或者在两端同时引入某种结构,如短语树结构或依存树结构。利用这些结构信息,基于句法的统计机器翻译就可以打破短

语系统的瓶颈限制:局限于局部上下文语境。一方面,在解码时,随着结构的不断递升,基于句法的模型能够控制的上下文范围越来越广,这些使得它处理远距离排序问题的能力也越来越强。另一方面,大量的结构标记和变量的引入,使得基于句法的模型能够处理非连续短语,并具有泛化能力。

从上面的介绍中还可以看到,各种基于句法的模型都有自己的优缺点。但是最终能否在机器翻译中显出明显的成效,要在三个方面打好根基。

1) 首先,模型在理论上应该确保解决机器翻译中的某些基本难点,或者简单地说,它是适合机器翻译的。举个例子说,同步上下文无关文法和同步 TAG 相比,从理论上说,后者应该更适合机器翻译,因为它能更好地解决非同构问题,而这个问题在机器翻译中又是普遍存在的。

2) 第二,模型应该尽可能简单、灵活、在计算上是可行的。这一点似乎和第一点有点矛盾,一般而言,由于机器翻译本身是一个非常困难的问题,如果模型太简单了,似乎就不能完美全面地解决某些问题。但是最终模型是通过计算机来实现的,因此计算的复杂性也是一个很重要的衡量标准。还拿 SCFG 和 STAG 来说,STAG 在理论上比 SCFG 完美,但是计算复杂性上大大高于 SCFG,这就是为什么目前基于 SCFG 的统计机器翻译系统要比基于 STAG 的系统要多的原因。再来对比一下树-树模型和树-串模型,理论上说,树-树模型由于应用了更多的句法结构知识,应该更适合于机器翻译,但是目前所看到的大部分基于树的系统还是基于树-串模型的,这是因为树-串模型相对而言要简单些,同时也具有很强的灵活性。

3) 最后,这些模型最终是要应用到实际的环境中的,所以工程技术也是一个很重要的因素。Och 本人也历次在 NIST 评测研讨会上不断强调工程的重要性。一个好的方法到底是不是有实际效果,要看最终的实现是不是很成功。很难想象充满了 Bug 的系统能够翻译出漂亮的译文。尽量缩短实验周期,不断进行错误分析,发现和修改系统存在的各种问题。一个模型的成功,除了它本身的优点之外,很多时候诸多的技术细节也决定了它最终的表现。这些都可以从实际的系统开发和研究中看得出来,如 Chiang 的系统能够有突破性的成功,除了层次性短语本身比一般短语更有优势之外,很重要的一点,他的系统本身融合了很多短语模型中证明成功的技术,如用 log-linear 模型集成多知识源,最小错误率

训练参数等。再比如,ISI的树-串模型能够最终取得好成绩,很重要的一点就是他们在搜索技术上花了很大的功夫^[36]。另外一些对统计机器翻译系统很重要的工程技术问题包括大规模语言模型、分布式计算等,这些在实际系统开发中都是要引起重视的。

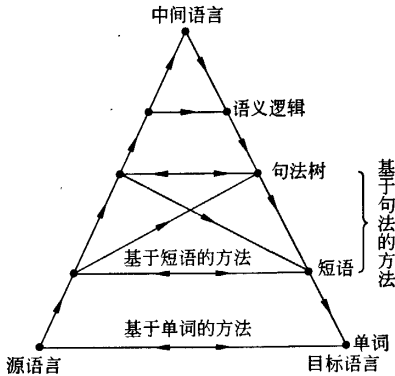


图3 统计机器翻译金字塔

仿照机器翻译中著名的金字塔结构,我们也可以构建一座统计机器翻译的金字塔^[37](见图3),以确定各种统计机器翻译方法所处的相对位置。在树-串方法中,可以看到有基于短语结构树的,也有基于依存树的,而在串到串和串到树方法中,可以看到目前的“树”基本上都是短语结构树,因此可以预测依存树和串之间的映射也是一种潜在的方法。统计机器翻译目前研究的重心已渐渐从String这一层上升到Tree这一层,以及这两层之间的互相转换上。在这两层之中,各种方法,组合情况正在得到不断的尝试和验证。这个金字塔结构不仅能够让我们很清晰地看到过去和现在统计机器翻译研究的情况,也可以帮我们预测未来统计机器翻译发展的方向,比如可能的方向包括从句法这一层上升到语义层,或在这两层之间建立某种联系。

参考文献:

- [1] Stuart M. Shieber and Yves Schabes. Synchronous Tree-adjoining Grammars [A]. In: Proceedings of the 13th COLING [C]. University of Helsinki, Helsinki, Finland; 1990.
- [2] Dekai Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora [J]. Computational Linguistics, 1997, 23:377-404.
- [3] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, Dragomir Radev. Final Report of Johns Hopkins 2003 Summer Workshop on Syntax for Statistical Machine Translation [R]. 2003.
- [4] [NIST] [http://www.nist.gov/speech/tests/mt/\[Z\]](http://www.nist.gov/speech/tests/mt/[Z]).
- [5] Colin Cherry and Dekang Lin. A Comparison of Syntactically Motivated Word Alignment Spaces [A]. In: Proceedings of the EACL 2006 [C]. Trento, Italy; 2006.
- [6] Fei Xia and Michael McCord. Improving a Statistical MT System with Automatically Learned Rewrite Patterns [A]. In: Proceedings of the COLING 2004 [C]. Geneva, Switzerland; 2004.
- [7] David Chiang. A Hierarchical Phrase-based Model for Statistical Machine Translation [A]. In: Proceedings of the ACL 2005 [C]. 2005. 263-270.
- [8] David Chiang. An Introduction to Synchronous CFGs [R]. 2005.
- [9] Dekai Wu and Hongsing Wong. Machine translation with a stochastic grammatical channel [A]. In: Proceedings of the ACL 1998 [C]. 1998.
- [10] Benjamin Wellington, Sonja Waxmonsky, I. Dan Melamed. Empirical Lower Bounds on the Complexity of Translational Equivalence [A]. In: Proceedings of the ACL 2006 [C]. Sydney, Australia; 2006.
- [11] R. Zens and H. Ney. A Comparative Study on Reordering Constraints in Statistical Machine Translation [A]. In: Proceedings of the ACL 2003 [C]. Sapporo, Japan; 2003. 144-151.
- [12] R. Zens, H. Ney, T. Watanabe and E. Sumita. Reordering Constraints for Phrase-Based Statistical Machine Translation [A]. In: Proceedings of the COLING 2004 [C]. Geneva, Switzerland; 2004. 205-211.
- [13] David Chiang. Hierarchical Phrase-based Translation [J]. Computational Linguistics 2007, 33 (2): 201-228.
- [14] F. J. Och, H. Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation [A]. In Proc. of ACL 2002 [C]. Philadelphia, PA; 2002. 295-302.
- [15] Hao Zhang, Liang Huang, Daniel Gildea and Kevin Knight. Synchronous Binarization for Machine Translation [A]. In: Proceedings of HLT/NAACL 2006 [C]. New York City, NY.; 2006.
- [16] Kenji Yamada and Kevin Knight. A Syntax-based Statistical Translation Model [A]. In: Proceedings of the ACL 2001 [C]. 2001.
- [17] Kenji Yamada and Kevin Knight. A Decoder for Syntax-Based Statistical MT [A]. In: Proc. of the ACL

- 2002 [C]. 2002.
- [18] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a Translation Rule? [A] In: Proceedings of the HLT-NAACL 2004 [C]. 273-280. 2004.
- [19] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. Scalable Inference and Training of Context-Rich Syntactic Models [A]. In: Proc. ACL-COLING 2006 [C]. 2006.
- [20] Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases [A]. In: Proceedings of EMNLP-2006 [C]. 2006.
- [21] Philipp Koehn, Franz Joseph Och, and Daniel Marcu. Statistical Phrase-Based Translation [A]. In: Proceedings of the HLT/NAACL 2003 [C]. 2003.
- [22] Liu Zhanyi, Wang Haifeng and Wu Hua. Example-based machine translation based on TSC and statistical generation [A]. In: Proceedings of the tenth Machine Translation Summit [C]. 2005. 25-32.
- [23] Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-String Alignment Template for Statistical Machine Translation [A]. In: Proceedings of the ACL 2006 [C]. 2006.
- [24] Liang Huang, Kevin Knight, and Aravind Joshi. Statistical Syntax-Directed Translation with Extended Domain of Locality [A]. In: Proceedings of the 7th AMTA [C]. Boston, MA; 2006.
- [25] Bonnie J. Dorr. 1994. Machine Translation Divergences: A Formal Description and Proposed Solution [J]. *Computational Linguistics*, 20(4): 597-633.
- [26] Jonathan Graehl and Kevin Knight. Training Tree Transducers [A]. In: Proceedings of NAACL 2004 [C]. 2004.
- [27] Daniel Gildea. Loosely Tree-based Alignment for Machine Translation [A]. In: Proceedings of the ACL 2003 [C]. Sapporo, Japan; 2003. 80- 87.
- [28] Aravind K. Joshi. 1987. An Introduction to Tree Adjoining Grammars [J]. In A. Manaster-Ramer, editor, *Mathematics of Language*. John Benjamins, Amsterdam.
- [29] Jason Eisner. Learning Non-isomorphic Tree Mappings for Machine Translation [A]. In: Proceedings of the ACL 2003 [C]. 2003.
- [30] Chris Quirk, Arul Menezes, and Colin Cherry. Dependency Tree Translation; Syntactically Informed Phrasal SMT [R]. Microsoft Research Technical Report; MSR-TR-2004-113. 2004.
- [31] Heidi J. Fox. Phrasal Cohesion and Statistical Machine Translation [A]. In: Proceedings of EMNLP 2002 [C]. 2002.
- [32] Lin Dekang. A Path-based Transfer Model for Machine Translation [A]. In: Proceedings of COLING 2004 [C]. 2004.
- [33] Chris Quirk, Arul Menezes and Colin Cherry. Dependency Treelet Translation; Syntactically Informed Phrasal SMT [A]. In: Proceedings of the ACL 2005 [C]. 2005.
- [34] Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation [A]. In: Proceedings of ACL 2003 [C]. 2003. 160-167.
- [35] Yuan Ding and Martha Palmer. Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars [A]. In: Proceedings of the ACL 2005 [C]. 2005.
- [36] Daniel Marcu. Argmax Search in Natural Language Processing [R]. Invited talk in ACL-COLING 2006. 2006.
- [37] Su, Keh-Yih. 2005. To Have Linguistic Tree Structures in Statistical Machine Translation? [A] In: Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE) [C]. Wuhan, China; October 30th-November 1st, 2005.