软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(5):1120−1131 [doi: 10.3724/SP.J.1001.2012.04055]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于中心/修饰依存重排序模型的短语 SMT*

刘 水[1+], 李 生[1], 赵铁军[1], 刘鹏远[2]

[1](哈尔滨工业大学 计算机科学与技术系,黑龙江 哈尔滨 150001)

[2](北京语言大学 应用语言学研究所,北京 100083)

Head-Modifier Dependency Reordering Model for Phrased SMT

LIU Shui[1+], LI Sheng[1], ZHAO Tie-Jun[1], LIU Peng-Yuan[2]

[1](Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](Applied Linguistics Research Institute of Beijing Language and Culture University, Beijing 100083, China)

+ Corresponding author: E-mail: water1981@gmail.com

Liu S, Li S, Zhao TJ, Liu PY. Head-Modifier dependency reordering model for phrased SMT. *Journal of Software*, 2012,23(5):1120−1131. http://www.jos.org.cn/1000-9825/4055.htm

Abstract: To enhance the reordering capacity of the phrase-based SMT (statistical machine translation), the study leverages the head-modifier dependency structure on the source to model the reordering. The model is added to baseline model in the form of soft-constraint way. The proposed model explores an approach to utilize the constituent based parse tree that the parse tree is mapped into sets of head-modifier relationships. Experimental results show that this model improves the local reordering significantly.

Key words: phrase-based SMT (statistical machine translation); reordering model; head-modifier relationship; non-lexicalized

摘 要: 为了提高基于短语的机器翻译系统的重排序能力,提出了一个基于源语言端的中心-修饰依存结构的重排序模型,并将该重排序模型以软约束的方式加入到机器翻译系统中.该排序模型提出了一种在机器翻译中应用句法树资源的方法,将句法树结构,通过将句法树映射成中心-修饰词的依存关系集合.该重排序模型在基于短语系统的默认参数设置下,显著地提升了系统的翻译质量.在系统原有的词汇化的重排序模型基础上,该重排序模型在翻译模型中融入了句法信息.实验结果显示,该模型可以明显地改善机器翻译系统的局部调序.

关键词: 短语机器翻译;重排序模型;中心修饰依存关系;无词汇化

中图法分类号: TP391 **文献标识码**: A

对于句法是否能够提升机器翻译的性能,在机器翻译领域曾经存在一些争论.一些研究认为,句法信息也许并没有办法提升机器翻译的性能.而最近越来越多的研究成果表明,句法信息可以有效地帮助机器翻译系统提高性能,并在许多关于机器翻译性能的评测中表现出优异的性能.

Chiang 在文献[1]中将基于句法的机器翻译模型分为形式化的(formal sense)和语言学的(linguistic sense)句法翻译模型.形式化的句法翻译模型在翻译过程中不应用任何语言学分析为辅助,在翻译规则的抽取过程中以

* 基金项目: 国家自然科学基金(60603032); 国家高技术研究发展计划(863)(2006AA010108)
收稿时间: 2010-07-01; 定稿时间: 2011-04-28

对齐作为其主要的限制条件.比如,在 Hiero[1],ITG[2],BTG[3]这些翻译模型中,翻译规则并不具备语言学的表现形式,只以(连续或者非连续的)短语形式存在.这些翻译中的句法规则,在翻译规则的抽取上仅对文法作简单的约束(比如对齐约束、长度约束、不连续短语中非终结符的数目等).

基于语言学的句法翻译模型采用具有语言学分析表现形式的翻译规则,这些规则通常需要符合一定语言学限制.在翻译过程中,解码器算法应用这些翻译规则,遍历或者形成某个基于语言学分析的句法结构,完全句法结构是此类翻译模型最常应用的语言学分析结构.目前,一些研究者在该方向上取得了很有价值的研究成果(Yamada[5],Liu[6],Zhang[7],Mi[8]).具有完全句法结构表现形式的翻译规则一方面使翻译模型具备了一定的语言学表达能力:一些复杂的翻译现象可以从完全句法分析的角度得到合理的解释,另一方面也为翻译模型带来了一些问题:首先,完全句法分析本身也是一个复杂的自然语言处理任务,目前还远远没有达到完美的程度,具有一定的错误率;其次,文献[9]的研究结果表明,某些不符合完全句法结构限制的短语可以显著地提高机器翻译的性能.为了解决上述问题,一些研究者提出了句法森林、N-BEST[8,10]等方法.通过这些方法,解码器扩大了最优路径的搜索空间,缓解了由于引入完全句法结构给机器翻译带来的问题.

文献[5]认为,机器翻译作为一个复杂的自然语言处理任务,对于翻译过程的更深入分析将产生更好的翻译结果.完全句法树是一种携带了丰富的结构信息的语言分析结果,可以有效地协助机器翻译过程提高翻译的质量.与以上将完全句法结构以翻译规则的方法融入翻译过程的方式不同,本文的重排序模型将源语言端的完全句法树结构以特征的形式融入到解码过程中,在对数线性模型下,以软约束(soft constraint)的方式辅助机器翻译过程.我们认为,软约束的引入模糊了以语言学分析为界限的机器翻译模型的分类方法,基于短语的翻译系统和基于形式化句法的翻译系统可以方便地引入语言学分析,从而在一定程度上弥补由于语言学分析的缺失而带来的不足.此外,语言学分析以特征的形式引入到翻译系统当中,也在一定程度上缓解了上文提到的由语言学分析带来的问题:语言学分析不再直接作用于翻译过程(翻译规则的抽取可以不遵循语言学分析的限制),而是以软约束的方式辅助机器翻译过程.在这个方向上,一些研究者已经取得了很有价值的研究成果:Chiang[1]和Marton[11]研究了层次短语规则中的违背/遵守句法结构边界的现象,Xiong[12]将句法树信息加入到 BTG 模型中,Zhang[13]将树核的信息加入到 BTG 模型中.这些研究成果表明,将语言学分析以软约束的方式加入到机器翻译系统中,可以有效地辅助翻译过程.

本文以一个基于短语的机器翻译系统 Moses 为基线,将基于完全句法树的特征以软约束的形式引入到解码过程中,明显地提升了基线模型局部重排序能力,并显著地提升了翻译质量.与其他基于软约束的工作相比,本文提出了句法树的一种新用法:将一棵句法树转化成若干个中心-修饰词的依存集合,进而本文定义了一个基于中心-修饰词的重排序模型,每个在源语言存在对齐的目标语言词都被该模型赋予一个重排序类型,在解码过程中,这些重排序类型的估计以特征的形式辅助机器翻译算法的解码过程.

1 基于短语的机器翻译模型

Moses[4]是目前最好的基于短语的机器翻译系统之一,在该模型翻译过程中,解码器进行基于 beam-search 的路径搜索,寻找满足公式(1)的翻译结果:

$$e_{best}=\arg\max_e p(e|f)p_{LM}(e)\omega^{length(e)} \quad (1)$$

其中,$p(e|f)$由短语翻译模型、扭曲重排序模型(distortion reordering 或者 monotonicity)和词汇化重排序模型组成,$p_{LM}(e)$是目标翻译的语言模型,$\omega^{length(e)}$是词惩罚模型.

在公式(1)的模型中,扭曲度重排序模型、词汇化重排序模型和语言模型对于翻译过程有重排序能力.扭曲重排序模型反映了一种粗略的源语言和目标语言之间的语序差异:几种不同语序的翻译可能具有相同的扭曲度,并且该重排序模型也并不是一个特化模型,并没有反映具体的重排序现象.可以认为,该模型实质上反映了一种泛化的翻译规律:在训练语料中,翻译模型对于翻译重排序的惩罚程度.语言模型是一个词汇的局部的重排序模型,该模型反映了目标语言中,词汇之间的共现关系.语言模型是该系统中一个重要的重排序手段,但由于是基于 N 元词汇共现的统计,该模型很难具备长距离的重排序能力:语言模型只能在局部某个范围内对翻译语

言重排序(这个范围取决于语言模型的元数).

词汇化重排序[14]是一个词汇化的重排序模型.该模型依照相邻翻译短语在源语言端的关系定义了 3 种翻译重排序类型.图 2 是该模型定义的翻译重排序的例子:翻译短语$\langle e_1\rangle$与翻译短语$\langle e_2,e_3\rangle$在左角线上相邻,定义为同序(monoton),简写为 m;翻译短语翻译$\langle e_4\rangle$与翻译短语$\langle e_5\rangle$在右对角线上相邻,定义为交换(swap),简写为 s;翻译短语$\langle e_5\rangle$和$\langle e_6\rangle$不相邻,定义为不连续(discontinuous),简写为 d.这个模型虽然可以对一些具体的重排序现象进行建模,而该模型只能估计翻译过程中两个相邻短语之间的重排序;在参数的估计过程中,该模型的重排序参数基于源语言端和目标语言端的词汇信息,采用了简单的极大似然估计,这些不利因素在一定程度上限制了该重排序模型的能力.

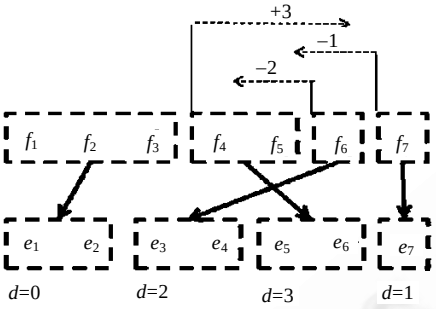


Fig.1 Distortion reordering model

图 1 扭曲度重排序模型

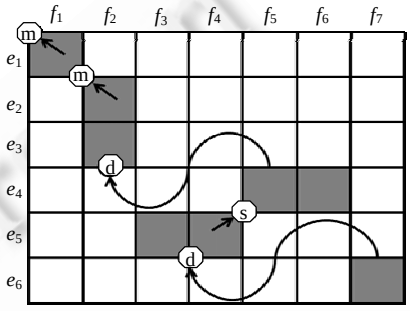


Fig.2 Lexicalized reordering model

图 2 词汇化重排序模型

2 基于中心-修饰依存结构的重排序模型

从以上对于 Moses 系统中的重排序模型可以看出,目前该系统中的重排序模型比较简单.也正是由于这个原因,在 Moses 的解码过程中,翻译的重排序被按照扭曲度(distortion)做了一定的限制,同序的翻译路径在一定程度上被鼓励.

与此同时,Moses 作为目前最好的短语翻译系统之一,在大规模语料上具有性能稳定、解码过程健壮的特点.因此,本文试图在不破坏其原有翻译算法的前提下,在源语言端加入句法信息,以增加其模型的重排序能力.

2.1 中心-修饰依存结构

本文的重排序模型将一棵基于 N 元树库文法(N-ary tree-bank grammar[15],简称 PCFG,以下简称为 N 元文法,也称为上下文无关文法)的完全句法树映射成中心-修饰依存结构的集合.一些研究[15,16]认为,N 元文法的方法有一些不适合于完全句法树的缺点:首先,N 元文法的概率估计比较粗略,而实际上,同一个 N 元文法的规则在句法树的不同位置的概率分布不尽相同;其次,N 元文法的规则不够灵活性,由于每个文法规则都由 N 个分析成分组成,因此文法规则的空间比较庞大,在训练集合中抽取的文法规则未必可以在训练结合中出现.基于以上考虑,在一些目前最好的完全句法分析方法[15,16]中,N 元文法规则通常被拆解成基于中心-修饰成分的依存结构.

此外,本文采用中心-修饰成分的依存结构还基于如下的考虑:首先,一些研究表明[17],依存结构在翻译中具有较好的一致性;其次,中心-修饰结构的是二元结构,与 N 元文法相比,在参数的估计方面受数据稀疏问题的影响更小;最后,这个结构是在句法分析中的成熟技术,本文在建模过程中可以方便地引入一些已经成熟的研究结果[15,18].

2.2 基于中心-修饰依存结构的重排序模型

图 3 给出的是源语言的完全句法树.图 4 中源语言端的结构是将其映射成一个中心-修饰依存的集合的实例.在该实例中,通过对于完全句法结构的转换,图 2 共形成 5 条依存弧.在同一条依存弧下,我们可以看到一些翻译规则中的重排序现象:在源语言端的“必须/VV”左依存于“做/VV”,其各自的翻译与源语言的顺序相同,“必须

/VV”的翻译“must”在“做/VV”的翻译“made”的左边;而“准备/NN”右依存于“做/VV”,其翻译顺序却与源语言顺序相反,“准备/NN”的翻译“preparations”在“做/VV”的翻译“made”的左边.我们认为,源语言端不同的上下文在一定程度上影响了翻译过程中的重排序.本文将翻译过程中的重排序情况分为两种:与目标语言具有相同相对语序的重排序 (以下简称为 m 语序);与目标语言具有不同相对语序的重排序(以下简称为 s 语序).

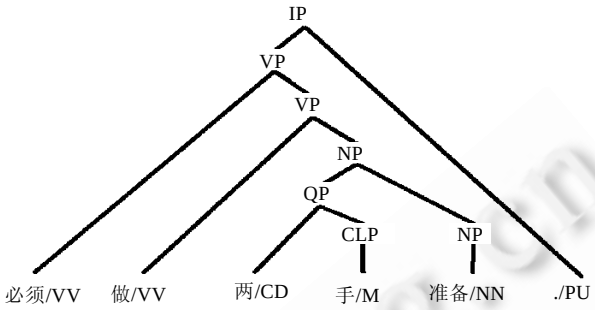


Fig.3 A full parse tree

图 3 完全句法分析树

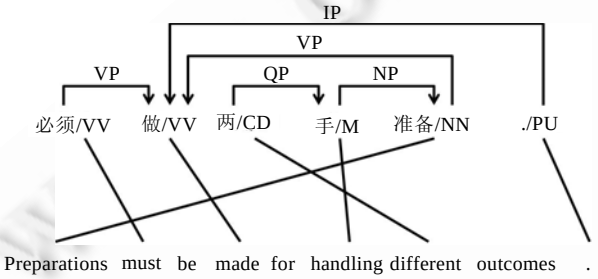


Fig.4 An head-modifier dependency relationship with alignment

图 4 带有对齐信息的中心-修饰关系

为了更好地阐述本文的重排序模型,本文首先给出机器翻译重排序模型中参照单元的概念:在机器翻译重重排序模型中,当前翻译单元的参照单元在本文中称为重排序模型的参照单元(下文简称为参照单元).基于这个定义,给出文中机器翻译重排序模型(该概念仅包含基于判别式的重排序模型)的定义:机器翻译的重排序模型是对某个翻译单元及其参照单元之间相对顺序进行建模的机器翻译模型.比如,ITG 重排序模型、BTG 重排序模型、扭曲度重排序模型选择短语作为翻译模型的重排序单元,并选择其相邻的翻译短语的作为重排序的参照单元.在文中,重排序模型的基本单元是词;在参照单元的选择上,选择该词的中心结点作为参照单元.这样选择参照单元的原因在于:本文是一种基于依存结构的重排序模型,在依存结构中,每个源语言词有多个孩子结点/修饰结点,而每个词最多有 1 个父/中心结点.如果将每个词的孩子结点选作重排序模型的参照单元就会使模型比较复杂,每个词的重排序将由许多结点决定;而将每个父亲结点选作重排序的参照单元显然可以在一定程度上简化重排序模型.

本文提出的翻译重排序模型与短语层面的重排序模型的一个最主要的不同之处在于:本文根据源语言的依存和对齐,选择词作为最小的重排序单元.在本文系统中,每个在源语言端有对齐的目标语言词都有一个重排序类型,进而将短语的重排序转化成词的重排序,使得重排序模型更为灵活.

下面给出本文重排序模型一个形式化的定义.给定源语言序列 $F=\langle f_1,f_2,\ldots,f_n\rangle$,其中,$f_n$ 表示语言序列中的第 n 个词;源语言端的依存结构 $D=\{(d(i),r(i))|1\leqslant i\leqslant n\}$,其中,$d(i)$表示在源语言词序列中,第 i 个词依存于第 $d(i)$个词,$r(i)$表示该依存弧对应的依存关系(下文将给出详细定义);目标语言序列 $E=\langle e_1,e_2,\ldots,e_m\rangle$,其中,$e_m$ 表示目标语言中的第 m 个词;源语言和目标语言之间的对齐关系 $A(i)$和 $A^{-1}(i)$,其中,$A(i)$表示源语言中的第 i 个词与目标语言

中的第 $A(i)$个词对齐,$A^{-1}(i)$表示目标语言中的第 i 个词与源语言中的第 $A^{-1}(i)$个词对齐;目标语言的重排序类型序列为 $O=\langle o_1,o_2,\ldots,o_m\rangle$,其中,$o_i\in\{m,s\}$.

2.2.1 翻译重排序中的重排序类型

在该模型中,选择源语言端每个词的中心结点作为重排序的锚点,则重排序类型 m 和 s,定义如下:

(1) 当 $A^{-1}(i)>d(A^{-1}(i)),i>A(d(A^{-1}(i)))$时,$o_i=m$;

(2) 当 $A^{-1}(i)<d(A^{-1}(i)),i<A(d(A^{-1}(i)))$时,$o_i=m$;

(3) 当 $o_i\neq m$ 时,$o_i=s$.

根据以上定义可知:在目标语言端,定义(1)为修饰结点在其中心结点的右侧,定义(2)为修饰结点在其中心结点的右侧;在目标语言端,在定义(1)与定义(2)中,修饰结点与其中心结点的翻译顺序保持不变,则定义为 m 类型重排序;其他情况则定义成 s 类型重排序.

例如在图 4 中,“must”的对齐词在源语言端为“必须/VV”,其中心结点“做/VV”对应的翻译为“made”,在源语言端“必须/VV”在“做/VV”的左侧,在目标语言端“must”同样在“made”的左侧,因此“must”的重排序类型为 m;同理可知,“Preparation”的重排序类型为 s.

2.2.2 重排序模型的参数估计

本文的重排序模型是对重排序序列 O 的一个估计:

$$p(O\mid F,E,D,A,A^{-1})=\prod_{i=1}^{m}p(o_i\mid r(i')) \tag{1}$$

其中,$i'=A^{-1}(i)$是 i 在源语言端对齐词的位置,m 是目标语言的长度,o_i 是每个目标语言的重排序类型.

从公式(1)可以看出,本文的重排序模型是作用在目标语言端的每个翻译词的重排序类型上的模型.在上式中, $r(i')$是源语言端第 i'个词与其中心结点的依存关系.该重排序关系 $r(\cdot)$作为影响重排序的上下文被加入到重排序概率参数的估计过程中.本文的依存重排序模型并没有采用既有的依存关系体系对 $r(\cdot)$进行定义,而是采用一些完全句法树和中心/修饰结构中的上下文对 $r(\cdot)$进行定义.通过基于中心/修饰依存结构的上下文的定义,本文的重排序模型更为灵活:依存关系不再局限于既有的依存关系定义(比如通常的主谓关系、修饰关系),而是可以根据系统性能的反馈进行定义,从而使重排序模型的参数估计过程变得更为灵活.下面给出 $r(\cdot)$详细的定义.

源语言端的中心/修饰依存 $r(\cdot)=\langle c_1,c_2,\ldots,c_n\rangle$,为每对中心/修饰依存关系的上下文集合.该集合中的元素 c_i $(1<i<n)$为依存结构中影响重排序的上下文,本文重排序模型的重排序参数以该上下文为条件估计出.因此,$r(\cdot)$的定义(包含哪些中心/修饰依存关系中的上下文)成为影响本文重排序模型中参数估计的一个关键.在 $r(\cdot)$的定义中,为了避免参数估计过程中词汇化对参数估计的影响,本文的重排序模型中并没有采用任何词汇化信息.本文在 $r(\cdot)$的定义借鉴了一些成熟的句法分析模型[16,18],采用一些其解码过程中估计参数的上下文,并根据实验效果的反馈对 $r(\cdot)$定义,其中包括:依存方向 *dir*、修饰结点的词性 *m_pos*、中心结点的词性 *h_pos*、父结点的句法标签 *label*、祖父结点的句法标签 *p_label*、依存结构中的临近的兄弟结点词性 *s_pos*.

表 1 为图 2 中源语言端的依存关系实例.从中可以看出,每个源语言端的词在本文都对应一个本文定义的依存关系.

Table 1 An example of head-modifier dependency relationship

表 1 中心修饰结构中的依存关系实例

f_i	$r(\cdot)$	*Dir*	*m_pos*	*h_pos*	*label*	*p_label*	*s_pos*
必须	$r(1)$	→	VV	VV	VP	IP	*NULL*
做	$r(2)$	↑	VV	TOP	TOP	TOP	*NULL*
两	$r(3)$	→	CD	M	QP	NP	*NULL*
手	$r(4)$	→	M	NN	NP	VP	*NULL*
准备	$r(5)$	←	NN	VV	VP	VP	*NULL*
.	$r(6)$	←	PU	VV	IP	TOP	NN

根据表 1,公式(1)的右部可以展开成如下形式:

$$p(o|r(\cdot))=p(o|dir,m\_pos,h\_pos,label,p\_label,s\_pos) \tag{2}$$

在上式的估计过程中,为了避免 0 概率的参数估计,本文在该参数估计的过程中引入加一平滑技术:

$$p(o\mid dir,m\_pos,h\_pos,label,p\_label,s\_pos)=\frac{F(o,dir,m\_pos,h\_pos,label,p\_label,s\_pos)+\alpha}{F(dir,m\_pos,h\_pos,label,p\_label,s\_pos)+2\times\alpha} \tag{3}$$

其中,$F(\cdot)$为统计事件在训练语料中出现的频率;α为加一平滑因子,本文采用下式估计:

$$\alpha=\frac{1}{C\times F(dir,m\_pos,h\_pos,label,p\_label,s\_pos)} \tag{4}$$

其中,C 为平滑常数,在本文实验中选取 C=5.

在公式(4)中,平滑因子α随统计空间的增大而变小,这个性质使本文的参数估计过程具有了一定的自适应特性:统计空间越大,其估计参数受到平滑因子的影响越小.这就在一定程度上缓解了数据稀疏对极大似然估计的影响.

3 对齐的前处理

本文通过源语言端的依存结构引入了一个词级的对齐模型:每个目标语言的词通过对齐结构映射到源语言中心-修饰结构中的一个依存关系,通过这个依存关系及其源语言端的依存对的翻译顺序,计算翻译的重排序.在实际翻译任务中,翻译现象的对齐情况比较复杂,多对一、一对多和空对齐的情况在语料中频繁出现.类似于文献[19]中的做法,本文在训练过程和解码过程之前对相关语料中的对齐结构进行了前处理.概括起来,本文的前处理主要为了达到以下目的:

- 简化模型:与文献[19]中前处理方法的效果相同,将多对一以及一对多的对齐结构转化为一对一的对齐结构,从而使本文的重排序模型可以简单地融入到解码过程当中.
- 模型需要:对齐结构需要满足在重排序模型中,确定重排序单元的重排序类型需求.每个源语言的词及其调序参照词的翻译需要有确定翻译位置(即在目标语言端存在与其对齐的词),以确定其重排序类型.本文为那些在目标语言中不存在翻译的源语言词找到一个对齐,以确定其翻译重排序类型和其修饰结点的重排序类型.

首先,给出本文对齐的表示:给定源语言序列 $F=\langle f_1,f_2,\ldots,f_n\rangle$,目标语言序列 $E=\langle e_1,e_2,\ldots,e_m\rangle$,若源语言中 f_i 与目标语言 e_j 存在对齐关系,则 $link(i,j)$=1;否则,$link(i,j)$=0.

本文对齐预处理的对象主要分为 3 类,基于以上对齐的定义,可以形式化表示为:在目标语言端有多个对齐词的源语言词的集合 $Link_{N\text{-}To\text{-}1}=\{f_i\mid f_i\in F,\sum_j link(i,j)>1\}$;在源语言端有多个对齐词的目标语言词的集合 $Link_{1\text{-}To\text{-}N}=\{e_i\mid e_i\in E,\sum_i link(i,j)>1\}$;在目标语言端不存在对齐的源语言词的集合 $Link_{NULL}=\{f_i\mid f_i\in F,\sum_j link(i,j)=0\}$.在以上分类中,对集合 $Link_{N\text{-}To\text{-}1}$ 和集合 $Link_{1\text{-}To\text{-}N}$ 进行前处理的目的在于简化模型;对集合 $Link_{NULL}$ 进行前处理的主要目的在于满足模型需要.

本文分别为以上的前处理对象定义了以下 3 种前处理操作:

- $Operation_d(f)$:其中,$f\in Link_{1\text{-}To\text{-}N}$;该操作根据词汇翻译概率,将仅保留词 f 在目标语言端的一个翻译词 e_m,即 $e_m=\arg\max_i P(e_i\mid f)link(e_i,f)$;丢弃 f 在目标语言端的其他对齐,令 $link(e_i,f)$=0,且 $i\neq m$.
- $Operation_b(f)$:其中,$f\in Link_{NULL}$;该操作为 f 在目标语言端寻找一个对齐词 e_i,在寻找这个对齐的词的过程中,我们应用一个启发式的规则:寻找距 f 最近的并且有对齐的源语言词,当在同一距离存在两个满足规则的源语言词时,优先选择左侧的词,并将其在目标语言端的对齐词 e_i 作为 f 的对齐词,即令 $link(e_i\mid f)$=1.
- $Operation_f(e)$:其中,$e\in Link_{N\text{-}To\text{-}1}$;该操作与 $Operation_d(f)$操作不同,并不会丢弃任何对齐结构,只是为 e 在源语言端寻找一个对齐的词 $f_n=\arg\max_i P(f_i\mid e)link(e,f_i)$,将该词作为 e 在解码和训练中确定重排序类型的对齐词.

通过以上定义的操作,本文算法将双语语料中的对齐结构转化为需要的形式.需要说明的是,本文的前处理是作用在同一个对齐结构上的,并且需要按照一定的顺序进行,如算法 1 所示.

算法 1. 对齐的前处理算法(alignment pre-processing algorithm).

输入:源语言序列 F 和目标语言序列 E.

输出:经过前处理的对齐结构.

1: **foreach** $f \in Link_{1\text{-To-}N}$ **do**
2: $Operation_d(f)$
3: **end for**
4: **foreach** $f \in Link_{NULL}$ **do**
5: $Operation_b(f)$
6: **end for**
7: **foreach** $e \in Link_{N\text{-To-}1}$ **do**
8: $Operation_f(e)$
9: **end for**

图 5 是一个对齐预处理算法的例子,图 5(a)中是一个复杂的对齐:源语言词 f_2 在目标语言端存对齐词 e_1 和 e_3,源语言词 f_4 在目标语言端不存在对齐词,目标语言词 e_1 在源语言端存在两个对齐词 f_1 和 f_2;图 5(b)应用 $Operation_d(f_2)$操作,根据翻译概率选择 e_3 作为 f_2 在目标语言端的对齐词,并丢弃 f_2 与 e_1 之间的对齐;图 5(c)应用 $Operation_b(f_3)$操作,将 e_2 作为 f_4 在目标语言端的对齐词,产生源语言端的对齐映射 $A(\cdot)$;图 5(d)应用 $Operation_f(e_2)$操作,选择 f_3 作为确定 e_2 重排序类型的源语言词,在目标语言端的对齐映射中丢弃 f_4 与 e_2 之间的对齐,产生目标语言端的对齐映射 $A^{-1}(\cdot)$.

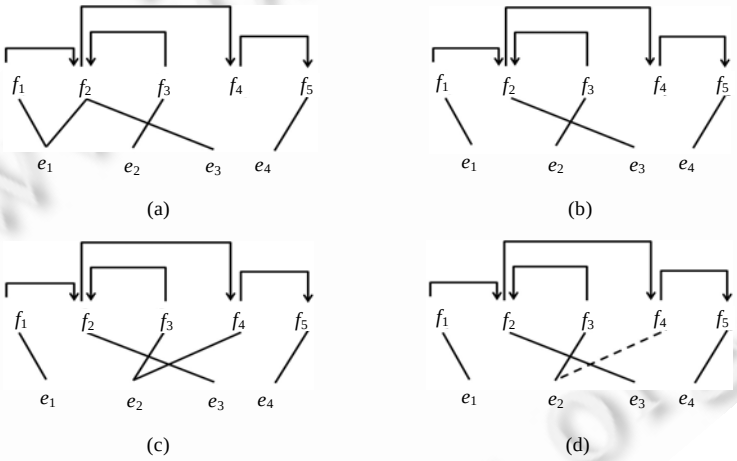


Fig.5 An example of alignment pre-processing

图 5 一个对齐的前处理的例子

4 模型的训练和解码

4.1 模型的训练

在训练本文的重排序模型之前,首先根据上文的前处理算法进行前处理,得到源语言和目标语言之间的对齐结构.然后,根据中心规则表[20]将源语言的完全句法分析树转化成依存结构.自左向右遍历目标语言端的词,并根据对齐、依存结构计算重排序类型,抽取公式(3)所需的统计事件.在此过程中,忽略那些在源语言端没有对齐的目标语言词.

4.2 模型的解码

本文的解码过程采用与文献[21]相同的解码算法.在解码过程中,该算法根据重排序限制,不断地在已有翻译的后面加入未被翻译的源语言的短语,直到所有的源语言词都被翻译.在该翻译过程中,已经翻译部分的语序在未来的翻译过程中保持不变:每个新翻译的部分将被加到已翻译部分的后面,因此,当前产生的翻译词的序号在未来的解码过程中也不会被改变(以下将该过程称为翻译的扩展).这个特点为本文的重排序模型加入到解码过程中提供了极大的方便:在翻译的扩展过程中,不必为每个新生成的翻译重新计算重排序得分,只需计算扩展部分的重排序得分.

为了计算新翻译短语的重排序得分,首先需要计算该重排序短语的重排序类型,根据上文对于重排序类型的定义,每个目标语言端的词的重排序类型的需要由以下信息确定:源语言端的依存关系、目标语言和源语言之间的对齐、源语言的翻译顺序(主要指中心结点和修饰结点之间的翻译顺序).目标语言端的依存关系可以通过对输入句子的句法分析得到,对齐可以被保存在短语表中,翻译顺序需要在解码过程中记录.为了记录该顺序,本文引入了一个 *bool* 类型的数组 *index* 来确定当前的翻译顺序:当源语言的第 i 个词被翻译时,*index* 中相应的第 i 个位置将被置为 *true*,否则其为 *false* 的状态.通过这个数据结构,本文可以计算在某个新扩展的短语中每个词的重排序类型.

算法 2. 解码中重排序类型的计算方法(algorithm of identifying the reordering types in decoding).

输入:翻译短语的目标语言和源语言的对齐函数 $A(\cdot)$和 $A^{-1}(\cdot)$;在源语言端的依存结构函数 $d(\cdot)$;源语言端的词索引 C,其中,$C[i]$=*true* 表示源语言中的第 i 个词被翻译;M 是翻译短语的目标语言端词的总数;S 为翻译短语在源语言端的对齐的起始位置,该数值用来将相对翻译位置转化为绝对翻译位置.

输出:翻译短语中目标语言端每个在源语言端有对齐的词的重排序类型.

```
1: for i=1,M do
2:     P←A^{-1}(i)+S
3:     if (d(P)<P) then
4:         if C[d(p)]=false then
5:             O[i]←m
6:         else
7:             O[i]←s
8:         end if
9:     else
10:        if C[d(p)]=true then
11:            O[i]←s
12:        else
13:            O[i]←m
14:        end if
15:    end if
16:    update_word_index( i );
17: end for
```

在以上算法中,函数 *update_word_index*(i)将目标语言中第 i 个词在源语言中对应的所有词的词索引置为 *true*.比如在如图 5(d)所示的对齐结构(经过前处理操作的对齐结构结构)中,当解码算法产生翻译词 e_2 以后,将会把 f_3 和 f_4 对应的词索引置为 *true*.

在翻译扩展过程中,通过以上算法,解码器可以计算出每个新扩展短语中词的翻译重排序类型,进而通过公式(1)计算出每个短语的重排序得分,将本文的重排序模型融入到翻译过程当中.

5 实验设置和讨论

5.1 实验设置

短语模型的抽取部分采用 LDC2003 的语料,该语料中包含 7.06 M 中文词和 9.15M 的英文词;NIST MT-02 作为参数学习的开发集;NIST MT-05 语料作为测试集;应用 Gigaword 语料中的 Xinhua 部分训练了一个四元的语言模型,该语料包含 181M 的英文单词.

为了得到本文的重排序模型的训练语料,首先使用 Stanford parser(2003)对 LDC2003 语料的中文部分进行完全句法分析;然后使用文献[20]中的中心规则将完全句法树转化为依存结构,并根据完全句法结构定义其依存关系.

在参数的学习的学习过程中,本文采用 MERT 训练工具进行训练,在训练过程中,本文的基线系统采用了如下的特征:

(1) 语言模型特征;

(2) 扭曲度重排序特征;

(3) 短语模型特征(详见文献[22],5 个特征);

(4) 词汇重排序模型特征(详见文献[14],6 个特征).

在本文的重排序模型中,在特征选择过程中,考虑到源语言部分依存方向对重排序得分分布的影响,本文根据源语言端的依存方向对重排序类型 m 和重排序类型 s 做了进一步的区分:将重排序类型 m 细分为源语言端为左依存类型(修饰词结点在中心结点的左侧)的 m^+重排序类型得分和右依存(修饰词结点在中心结点的右侧)类型的 m^-重排序类型得分;类似地,将重排序类型 s 分为 s^+和 s^-.因此,本文的重排序模型在基线系统中加入以下的特征组:

(5) 4 分类重排序类型得分(4 个特征,分别为 m^+,m^-,s^+和 s^-重排序类型得分);

(6) 4 分类重排序类型分别的个数(4 个特征,分别为 m^+,m^-,s^+和 s^-重排序类型的个数).

在解码过程中,系统的默认设置:翻译栈大小为 200,beam-search 剪枝限为 1/100000,每个源语言的跨度(span)最多载入 50 个翻译短语,扭曲度的最大限制为 6.

5.2 实验结果及讨论

表 2 为本文算法在 LDC2003 中抽取的事件的一个简单统计,第 2 列是源语言端为左依存结构的重排序分布,第 3 列为右依存结构的重排序分类.从表 2 中可以看出,重排序类型 m 占据了整个统计空间的大约 3/4,重排序类型 s 大约只占 1/4.也就是说,从本文依存结构重排序的观点上看,在本文的翻译语料中,多数的翻译过程并不改变翻译顺序:中英文之间的语序差异并不十分明显.这在一定程度上解释了:一些在翻译重排序的能力上存在一定局限性的翻译模型[2,3,22],在中英文翻译的翻译任务中仍然能够达到一定的性能.

Table 2 Distribution of reordering types

表 2 重排序类型分布

	+ (%)	− (%)	总数(%)
m	27.61	47.75	75.36
s	3.69	20.94	24.63

表 3 为本系统在默认参数下的性能,其中,baseline 模型在解码过程中采用特征组 1,2,3;baseline$_{rm}$ 模型在解码过程中采用特征组 1,2,3,4;H-M 模型在解码过程中采用特征组 1,2,3,5;H-M$_{rm}$ 模型在解码过程中采用特征组 1,2,3,4,5.

从表 3 中可以看出,在默认参数下,加入本文的翻译模型后,翻译系统的性能参数有显著的提升.在默认参数下,本文的重排序模型在系统性能上优于词汇化重排序模型.由于基线系统的全局重排序能力有限,在默认的扭曲度限制下,该限制保证基线系统的重排序只是在一定的范围内.这样做虽然限制了模型的翻译能力,但在一定程度上保证了系统性能.为了进一步研究本文模型的全局重排序能力,本文增大了扭曲度限制的范围,见表 4.

Table 3 Performance on default setting

表 3 默认设置下的性能

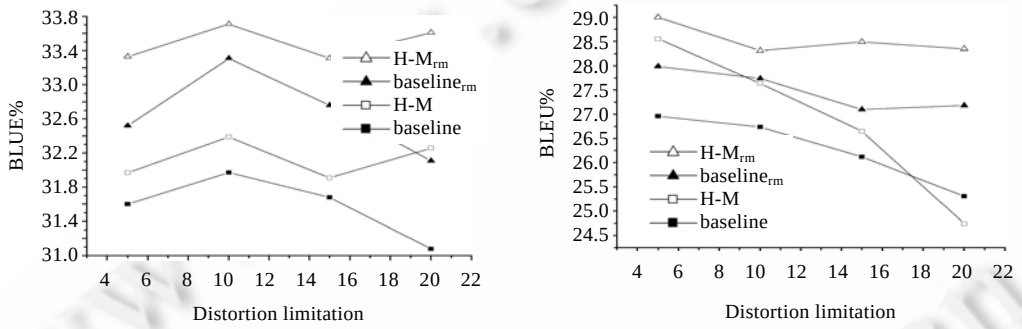
	BLEU%	NIST
Baseline	27.06	7.7898
$baseline_{rm}$	27.58	7.8477
H-M	28.47	8.1491
$H\text{-}M_{rm}$	29.06	8.0875

Table 4 Performance of different distortion limit settings

表 4 不同扭曲度限制设置的性能

Dis-Limitation	5		10		15		20	
corpus	NIST-02	NIST-05	NIST-02	NIST-05	NIST-02	NIST-05	NIST-02	NIST-05
baseline	31.60	26.96	31.97	26.74	31.68	26.12	31.08	25.31
$baseline_{rm}$	32.52	27.99	33.31	27.74	32.76	27.10	32.11	27.18
H-M	31.97	28.55	32.39	27.64	31.91	26.65	32.26	24.74
$H\text{-}M_{rm}$	33.33	29.00	33.71	28.31	33.31	28.49	33.61	28.35

从表 4 和图 6 中可以看出:随着重排序限制的增大,H-M 和 baseline 系统的性能随着扭曲度限制的增加在抽样点上性能下降得十分明显;$baseline_{rm}$ 和 $H\text{-}M_{rm}$ 的系统性能下降略为平缓.我们认为,$baseline_{rm}$ 性能优于 baseline,$H\text{-}M_{rm}$ 的性能优于 H-M,说明词汇化重排序可以提升短语翻译系统的性能.



(a) 开发集(NIST-02) (b) 测试集(NIST-05)

Fig.6 Performance on different settings of distortion limitation

图 6 不同扭曲度剪枝设置上的系统性能

由于 baseline,$baseline_{rm}$ 系统的重排序能力存在一定的局限性,将扭曲度的限制在一定范围内可以在一定程度上保证这些系统的翻译质量.单独使用本文的重排序模型的 H-M 系统并没有明显地改善这个问题,在几个抽样点上,系统性能波动也很明显.我们认为,这是由于本文的重排序模型是作用在源语言端的中心-修饰结构的重排序模型,只要中心-修饰词的翻译顺序不变,其重排序得分就不会有变化,多种翻译重排序现象可能对应一个重排序得分,长距离重排序能力有限.因此,随着扭曲度剪枝限制的扩大,系统性能下降得很快.

与只采用词汇化重排序模型的 $H\text{-}M_{rm}$ 相比,同时使用词汇化重排序和本文重排序模型的系统 $H\text{-}M_{rm}$ 在几个抽样点上性能得到了不同程度的提升.我们认为,这是由于与词汇化重排序模型相比,本文的模型在重排序时没有采用词汇上下文特征,而是采用非词汇化的句法上下文特征,这两种重排序模型在一定程度上存在重排序上的互补关系.

本文的模型明显地改善了短语翻译系统的局部重排序能力.在基于短语的机器翻译系统中,将扭曲度限制在一定范围内,有利于系统性能.

5.3 相关方法

本文的重排序模型将源语言的基于中心/修饰关系的依存结构以软约束的形式引入到翻译过程,一些研究者曾在翻译过程中引入依存结构信息.Ding[23]在翻译模型中同时引入了目标语言依存树和源语言依存树.该模

型中的翻译规则是从两端依存树中抽取的依存稚树(treelet),通过粘接和替换操作,该模型将目标语言和源语言中的依存结构引入到翻译过程中.该语法的表达能力可被看成是一种弱等价形式的 CFG 文法.Lin[24]应用源语言的依存结构定义了以源语言端依存结构为限制的路径,该模型在训练时抽取所有可以组成路径依存及其对应的目标语言片段,在解码时,该方法遍历依存树,产生目标语言翻译.Quirk[25]在 Lin[24]的模型基础上,将依存路径扩展成更复杂的稚树.Wang[26]通过源语言端的依存关系,在翻译之前首先对源语言端的短语进行重排序,然后对重排序后的源语言进行翻译.Shen[27]在层次短语的框架下提出了一个基于目标语言端依存结构的翻译模型,该模型可被看作一个串-树模型,每个层次短语中的规则在抽取时都保存源语言端的依存结构,并要符合其对于依存结构的限制.在翻译中,解码算法应用定义的操作将目标语言端的依存结构拼接成依存结构,并计算其依存概率(可被看作一种依存语言模型),作为特征融入到翻译过程中.

本文的重排序模型与以上基于依存结构的翻译模型都有区别:首先,文献[23−26]将依存结构以显示句法的形式融入到翻译过程中,而本文的模型将依存结构以特征的形式融入到短语翻译系统中,本文的重排序模型与Shen[27]提出的模型同属于这个类型的翻译模型;其次,本文的模型与 Shen[27]的模型在对原翻译模型产生影响上存在一个明显的不同,Shen[27]的模型需要根据目标语言端的依存结构对翻译规则进行限制,以使解码过程适应新的翻译模型(文献[27]中显示,这种做法并未提升翻译质量,反而使性能略有降低),本文的重排序模型不需要对原有模型的翻译规则进行任何限制,从而避免了由于翻译规则的减少对翻译性能产生的影响.

6 结论及展望

本文在目前最好的短语翻译系统中以软约束的方法引入了一个基于完全句法树的中心-修饰结构的重排序模型.该重排序模型提出了完全句法树的一种新用法,将句法树映射成基于句法上下文的中心-修饰依存结构.本文的重排序模型与词汇化模型存在一定的互补关系,同时使用这两个模型的性能显著高于使用单一系统的性能,并明显地降低了翻译系统对扭曲度限制的的敏感程度,使翻译系统的性能稳定在一定的范围之内.

致谢 在此,我们向对本文的工作给予支持和建议的实验中心老师,尤其是张民老师、徐志明教授、曹海龙博士及哈尔滨工业大学机器翻译实验室的其他老师和同学表示感谢.

References:

[1] Chiang D. A hierarchical phrase-based model for SMT. In: Proc. of the ACL 2005. 2005. 263−270.

[2] Wu DK. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Lingustics, 1997, 23(3):377−403.

[3] Xiong DY, Liu Q, Lin SX. Maximum entropy based phrase reordering model for statistical machine translation. In: Proc. of the COLING-ACL 2006. 2006. [doi: 10.3115/1220175.1220241]

[4] Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen WD, Moran C, Zens R,Dyer C, Bojar O, Constantin A, Herbst E. Moses: Open source toolkit for statistical machine translation. In: Proc. of the ACL 2007. 2007.

[5] Yamada K, Knight K. A syntax-based statistical translation model. In: Proc. of the ACL 2001. 2001. 523−530. [doi: 10.3115/1073012.1073079]

[6] Liu Y, Liu Q, Lin SX. Tree-to-String alignment template for statistical machine translation. In: Proc. of the ACL 2006. 2006. [doi: 10.3115/1220175.1220252]

[7] Zhang M, Jiang HF, Aw AT, Li HZ, Tan CL, Li S. A tree sequence alignment-based tree-to-tree translation model. In: Proc. of the ACL-HLT 2008. 2008. 559−567.

[8] Mi HT, Huang L. Forest-Based translation rule extraction. In: Proc. of the ENMLP 2008. 2008. 206−214.

[9] DeNeefe S, Knight K, Wang W, Marcu D. What can syntax-based MT learn from phrase-based MT? In: Proc. of the EMNLP-CoNULL 2007. 2007.

[10] Zhang H, Zhang M, Li HZ, Aw A, Tan CL. Forest-Based tree sequence to string translation model. In: Proc. of the ACL 2009. 2009. 172−180. [doi: 10.3115/1687878.1687904]

[11] Marton Y, Resnik P. Soft syntactic constraints for hierarchical phrased-based translation. In: Proc. of the ACL 2008. 2008. 1003−1011.

[12] Xiong DY, Zhang M, Aw A, Li HZ. A syntax-driven bracketing model for phrase-based translation. In: Proc. of the ACL 2009. 2009. 315−323.

[13] Zhang M, Li HZ. Tree kernel-based SVM with structured syntactic knowledge for BTG-based phrase reordering. In: Proc. of the 2009 Conf. on Empirical Methods in Natural Language Processing. 2009. 698−707.

[14] Koehn P, Axelrod A, Birch Mayne A, Callison-Burch C, Osborne M, Talbot D. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In: Proc. of the Int'l Workshop on Spoken Language Translation. 2005.

[15] Klein D, Manning CD. Accurate unlexicalized parsing. In: Proc. of the ACL 2003. 2003. 423−430. [doi: 10.3115/1075096.1075150]

[16] Collins MJ, Marcus MP. Head-Driven statistical models for natural language parsing [Ph.D. Thesis]. University of Pennsylvania, 1999.

[17] Fox HJ. Phrasal cohesion and statistical machine translation. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. 2002. [doi: 10.3115/1118693.1118732]

[18] Johnson M. PCFG models of linguistic tree representations. Computational Linguistics, 1998,24(4):613−632.

[19] Quirk C, Menezes A, Cherry C. Dependency treelet translation: syntactically informed phrasal SMT. In: Proc. of the ACL 2005. 2005. 271−279. [doi: 10.3115/1219840.1219874]

[20] Bikel DM. On the parameter space of generative lexicalized statistical parsing models [Ph.D. Thesis]. University of Pennsylvania, 2004.

[21] Koehn P. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In: Proc. of the AMTA 2004. Washington, 2004. [doi: 10.1007/978-3-540-30194-3_13]

[22] Koehn P, Och FJ, Marcu D. Statistical phrase-based translation. In: Proc. of the HLT-NAACL 2003. 2003. [doi: 10.3115/1073445.1073462]

[23] Ding Y, Palmer M. Machine translation using probabilistic synchronous dependency insertion grammars. In: Proc. of the ACL 2005. 2005. 541−548.

[24] Lin DK. A path-based transfer model for machine translation. In: Proc. of the COLING 2004. 2004. [doi: 10.3115/1220355.1220445]

[25] Quirk C, Menezes A, Cherry C. Dependency treelet translation: Syntactically informed phrasal SMT. In: Proc. of the ACL 2005. Ann Arbor, 2005. 271−279. [doi: 10.3115/1219840.1219874]

[26] Chao W, Collins M, Koehn P. Chinese syntactic reordering for statistical machine translation. In: Proc. of the EMNLP 2007. Prague, 2007. 737−745.

[27] Shen LB, Xu JX, Weischedel R. A new string-to-dependency machine translation algorithm with a target dependency language model. In: Proc. of the ACL 2008. Columbus, 2008. 577−585.

刘水(1981－),男,吉林省吉林市人,博士生,主要研究领域为句法分析,机器翻译.

赵铁军(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为自然语言处理,机器翻译.

李生(1943－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为人工智能,自然语言处理.

刘鹏远(1974－),男,博士,CCF 会员,主要研究领域为词义消岐,机器翻译.