利用句法信息改进交互式机器翻译

张亚鹏，叶娜，蔡东风，刘瑞倩

（沈阳航空航天大学 知识工程研究中心，辽宁 沈阳 110136）

摘要：当前的自动机器翻译还远远没有达到我们的预期。要想获得直接可用的译文，往往需要翻译人员对自动翻译系统输出进行后处理。在交互式机器翻译的框架内，翻译系统和用户协同工作，共同完成翻译任务。本文利用基于短语模型的机器翻译系统，建立交互式翻译系统。所以，该交互式机器翻译系统，会存在传统的翻译系统固有的问题，即基于短语模型的翻译系统的调序模型过于简单。针对短语模型中比较简单的调序模型，本文提出利用句法信息，子树信息，指导翻译假设扩展。并结合交互式机器翻译中的特殊特征译文前缀，共同完成翻译。实验表明，该方法可以有效的减少用户的工作量。

关键词：交互式机器翻译；调序模型；子树信息；译文前缀

中图分类号：TP391　　**文献标识码：**A

Using Syntactic Information to Improve Interactive Machine Translation

ZHANG Yapeng，YE Na，CAI Dongfeng，LIU Ruiqian

（Research Center for Knowledge Engineering of SAU，Shenyang，Liaoning 110136，china）

Abstract: Current automatic machine translation still far from the goal we propose. In order to generate error-free translations and human intervention is often required to correct their output, in the framework of an interactive machine translation, additionally, translation systems and users are usually work together to complete the translation task. Here, we describe the use of phrase-based model machine translation systems which establish an interactive translation system framework. After all, Because of traditional phrase-based machine translation system still have inherent problem, so even if we use interactive information, there is still phrase reordering model based translation system problem which is too simple. However, With regard to phrase model were so simple in reordering, we propose the use of syntactic information, which is subtree information that direct the translation assumption extend. Combined with translation prefix which as the interactive machine translation special feature, we finally accomplish this method of translation system. Experimental results show that this method can effectively reduce user workload.

Key words: interactive machine translation; reordering model; subtree information; translation prefix

1 引言

尽管机器翻译在最近的几十年取得了很大的进展，但是，现有的自动机器翻译系统，只是在有限的领域里，可以输出直接可用的高质量的译文。对于大部分领域，用户所需要的直接可用的译文，都必须由拥有翻译知识的译员，对机器翻译系统输出的译文进行后处理，然后才能推送给用户。在这种模式下，译员可以利用翻译系统推送的译文完成翻译任务，但是，机翻译系统却不能利用译员的翻译知识。于是，一些研究人员提出了交互式机器翻译框架，在此框架内，允许译员人工干预翻译过程。首先机器翻译系

基金项目：国家科技支撑计划（2012BAH14F05）

作者简介：张亚鹏（1988—），男，硕士研究生，主要研究方向为交互式机器翻译；刘瑞倩（1988—），女，硕士研究生，主要研究方向机器翻译；叶娜（1981—），女，博士，讲师，主要研究方向为辅助翻译、文本挖掘；蔡东风，男，博士，教授，主要研究方向为人工智能、自然语言处理。

统会对给定的待翻译句子推送出可能的译文，然后译员可以对翻译系统推送出的译文做出接受、修改或舍弃等操作，然后机器翻译系统会根据译员当前的操作做出下一步的预测，循环进行此过程，直到译员得到最终想要的译文。图 1 展示了一个经典的交互式机器翻译过程。

在这里我们要将一个汉语句子（source）“任何 不 属 客船 的 船舶 。”翻译为英文译文（reference）“Any ship other than a passenger ship .”。在开始交互之前（interaction-0），系统首先推荐一个可能的译文（或译文后缀，t_s）。在第一次交互（interaction-1）中，用户挪动光标来接受译文的前 4 个字符“Any ”（空格也包含在内），并且用键盘输入字符 s（k），然后系统根据用户修改后的译文前缀立即给出新的译文后缀“hip other than passenger ships .”。第二次交互（interaction-2）的境况类似。在最后一次交互时用户完全接受了系统当前推荐的译文。

Source(ch)：任何 不 属 客船 的 船舶 。
Reference(en)：Any ship other than a passenger ship .

interaction-0	t_p k t_s	Any of ships other than passenger ships .
interaction-1	t_p k t_s	Any **s** hip other than passenger ships .
interaction-2	t_p k t_s	Any ship other than **a** passenger ship .
acceptance	t_p	Any ship other than a passenger ship .

图 1 交互式翻译实例

关于交互式翻译系统的解码，与传统的完全自动机器翻译的解码原理是一样的，因此交互式的机器翻译翻译系统可以采用基于栈的解码策略，利用多栈或者是柱搜索解码算法。不同的地方在于，在交互式翻译系统解码时，会考察当前的翻译假设是否符合译文前缀，若不符合译文前缀则不加入到待扩展假设中。然后一步步扩展，生成最终译文。

本文在基于短语的交互式机器翻译的基础上，建立交互式机器翻译框架，针对传统的基于短语的翻译系统固有的问题，调序模型过于简单，提出了利用句法信息，子树信息，指导翻译假设扩展。并且结合翻译人员给予的译文前缀，相比于传统的机器翻译系统，交互式机器翻译系统的特殊特征，用三种策略把子树信息加入到交互式机器翻译系统的解码当中。第一种：只在完全匹配译员译文前缀之前的翻译假设扩展时，使用子树信息指导翻译假设的扩展；第二种：只在完全匹配译文前缀之后的翻译假设的扩展时使用子树信息作为指导；第三种：在整个翻译假设扩展当中都使用子树信息进行指导。实验结果表明，三种策略相比于基线系统，都能减少译员的工作量，但是第三种策略变现效果最好。

本文结构安排如下：在第二部分，介绍与本文相关的研究；子树信息的抽取及如何把子树信息嵌入到机器翻译系统的解码当中将在第三部分详细介绍；第四部分会介绍实验配置、实验结果及分析；第五部分对本文进行总结及以后的工作设想。

2 相关工作

在这部分中，介绍一些交互式机器翻译方面的其他研究人员的工作。

早期的交互式机器翻译研究，研究人员

主要的研究点集中在对源语言文本的解释和消歧。Foster 在 1997 年提出了 TransType 的基本系统[1]，该系统第一次将交互式机器翻译的关注点从对源语言文本的解释分析转移到目标语言文本的生成上，减轻了译员的工作负担提高了效率，并且使译员可以控制翻译系统输出的译文。之后的几年当中，又有很多的研究人员对 TransType 系统进行了改进。Langlais 等人在 2000 年对系统的用户界面和词的预测提出了改进[2]。2002 年，由许多欧盟研究机构共同参与的 TransType2 项目，创新性的把一个完全的基于数据驱动的机器翻译系统嵌入到交互式翻译框架中，并且在每一次的交互过程中，翻译系统都会根据翻译人员给出的译文前缀，预测出一个或者多个最好的后缀补全译文，供翻译人员选择。在 TransType2 项目中，很多的研究人员对系统进行分析，并且提出很多种方法来解决这些问题。TransType 的这两个项目极大的推动了交互式机器翻译技术的发展。2010 年，Ortiz 和 Casacuberta 等人，将在线学习的思想加入到了交互式机器翻译技术当中。其主要思想是利用用户的反馈信息来不断的完善系统的底层模型[3]。González-Rubio 和 Ortiz 等人，将机器译文的置信度评价作为其是否需要和翻译人员进行交互的衡量，从而有效地平衡了翻译人员的工作量和系统翻译结果的准确率[4]。2012 年，González-Rubio 和 Ortiz 等人[5]，将动态学习的方法引入到交互式机器翻译系统当中，使系统可以增量式的从已经翻译完的句子中学习，从而明显地提高后续句子的翻译准确率，有效减少了翻译人员的工作量。2013 年，Jesús González-Rubio 和 Daniel Ortíz-Martinez 等人[6]，将基于层次短语的翻译模型应用到了交互式机器翻译当中，并且采用了超图作为机器和用户之间的交互接口。

在之前的研究中，研究人员从对源语言的分析转移到对目标语言的生成，并且把在线学习和动态学习的思想加入到模型中，但都没有使用句法信息对翻译系统进行改进。

3 子树信息抽取及嵌入

这一部分，主要讲述子树信息的抽取及如何将子树信息嵌入到翻译系统的解码中。

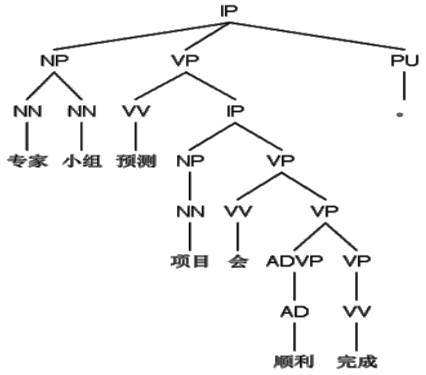

图 2 短语结构树的示例

3.1 子树信息抽取

```
Input: a data struct for a tree
Output: the subtree in this tree
Auxiliar: subtree (define a data struct to store the subtree)
      GetValidateSubtree() (return a flag for jude the span of a sentence is a legal subtree or not)
Begin
   string flag
   int i，j
      for i = 0,…n-2 do
            for j = 1,..n-1 do
            flag = GetValidateSubtree（i ， j）
                  if flag is legal
                  subtree.insert（flag ， i ， j）
            end for
      end for
   return subtree
End
```

图 3 抽取子树伪代码

句法树采用短语结构树。该句法结构把句子细分成更小的单位，然后通过短语连接起来。

子树是一个句子中相对独立的一部分，它可以是一个名词短语或动词短语。我们使用的子树信息，它是一个三元组，如公式（1）所示，它至少包含有句子的两个词,我们所用

到的子树并不包含整棵句法树，因为整棵句法树在本文中无任何意义。

$$S = (N, In\_s, In\_e) \quad (1)$$

N 表示子树名称，In_s 表示子树开始词在句子中的位置，In_e 表示子树结束词在句子中的位置。

系统得到待翻译的句子之后，我们首先用句法分析器对句子进行句法分析，生成短语结构句法树，如图 2 所示。经图 3 的伪代码处理之后，我们得到句子的子树信息。最终我们得到的子树是（NP , 0 , 1）、（VP , 2 , 6）、（IP , 3 , 6）、（VP , 4 , 6）、（VP , 5 , 6）。

3.2 子树信息的嵌入

对句子进行翻译时，应该翻译完成一个子树之后，才能对其他子树进行翻译，我们就把这个原则加入到交互式翻译系统框架中。本文中，我们使用基于短语的交互式翻译系统框架，利用多栈解码算法对短语系统进行解码。在每个代表当前翻译假设覆盖源语言词个数的大栈中，有很多覆盖不同位置但覆盖源语言词个数的小栈。当扩展翻译假设时，我们会选取每个大栈里的每个小栈中最大分值的翻译假设进行扩展。在这里我们使用子树信息选择更合适的翻译假设，由于短语扩展存在调序现象，所以覆盖相同源语言词的翻译假设可能是由不同的短语组成的，选取短语假设扩展时，在覆盖源语言词个数且源语言词位置相同的多个翻译假设中，若存在符合子树限制的翻译假设，则选择此翻译假设进行扩展，若不存在，我们按照传统的翻译假设选择方法，选择翻译假设进行扩展，当出现多个符合子树限制翻译假设时，我们选择分值最高的那个翻译假设进行扩展。

符合子树限制的定义是：当前翻译假设包含的上一个被翻译的短语和最后一个被翻译的短语所包含的词在同一个子树内。为了更好的结合基于短语的翻译模型，若当前所选择的子树只有一个连续的短语未被翻译且这个连续的短语在子树的边界上，允许扩展当前子树未包含的源语言词，前提是，当前所扩展的短语完全包含当前子树未翻译的词。

在判断翻译假设是否符合子树限制时，只使用翻译假设的上一个被翻译的短语和当前被翻译的短语是一种软策略，考虑到句法分析的性能，我们并不要求翻译假设的每一次扩展都符合子树限制，这样能够更好的利用原有系统短语扩展的优势。

然后结合交互式翻译所特有的特征——译文前缀，本文提出三种策略，第一种：只把子树信息应用到当前所选择的翻译假设未覆盖译文前缀时；第二种：只把子树信息应用到当前所选的翻译假设覆盖译文前缀之后；第三种：把前面两种结合起来，在整个句子的翻译中使用子树信息。三种策略的伪代码如图 4、5、6 所示。

```
Input:small_stack(the small stack which contains much hypothesis that covered same source words)
Output:hyp(a hypothesis to be extended)
Auxiliar:prefix(user validated prefix)    Subtree_information(the subtrees for current sentence)
Begin
      hyp = GetMaxHypothesis(small_stack,prefix)
   if hyp.FullCoveredPrefix == false and hyp.empty ==false
       hyp = GetMaxHypothesis(small_stack,prefix,subtree_information)
   return hyp
End
```

图 4 只在所选翻译假设未覆盖翻译前缀时使用子树信息伪代码

```
Input:small_stack(the small stack which contains much hypothesis that covered same source words)
Output:hyp(a hypothesis to be extended)
Auxiliar:prefix(user validated prefix)    Subtree_information(the subtrees for current sentence)
Begin
      hyp = GetMaxHypothesis(small_stack,prefix)
   if hyp.FullCoveredPrefix ==true and hyp.empty ==false
       hyp = GetMaxHypothesis(small_stack,prefix,subtree_information)
   return hyp
End
```

图 5 只在所选翻译假设已经覆盖翻译前缀时使用子树信息伪代码

```
Input:small_stack(the small stack which contains much hypothesis that covered same source words)
Output:hyp(a hypothesis to be extended)
Auxiliar:prefix(user validated prefix) Subtree_information(the subtrees for current sentence)
Begin
       hyp = GetMaxHypothesis(small_stack,prefix,subtree_information)
   return hyp
End
```

图 6 在整个翻译过程中使用子树信息伪代码

子树抽取时，抽取子树之间会存在嵌套且仅仅对包含整个句子的子树的特殊子树限制抽取。会造成包含句子词的个数过多的情况出现，这将导致翻译假设对子树限制不敏感。针对子树嵌套的情况，根据子树包含句子中词的个数，我们提出了最大子树策略（max_subtree）和最小子树策略（min_subtree），当出现子树嵌套情况时，根据策略不同，选取不同的子树。为了避免出现包含句子中词个数过多的子树出现，我们通过子树包含词的个数与整个句子词的个数的比值（RatioSubtreeSentece）对所抽取子树进行过滤。另外，在选取符合子树限制的翻译假设进行扩展时，我们还应该考虑当前所选择的翻译假设的分值与传统的所选择的翻译假设的分值进行比较，对于一些分值过低的但是符合子树限制的翻译假设进行舍弃，因为如果分值过低，在下一步的剪枝策略时也会被舍弃。我们把这个因素定义为分值比（score_ratio），在实验环节，会对以上提出的可能影响到系统性能的参数进行单独实验。

4 实验设置及结果分析

在这一部分，对实验语料的信息、评价标准和实验结果进行描述，并对实验结果进行分析。

4.1 语料信息

我们的实验采用部分的汉英平行语料 Hong Kong Laws Parallel Text（LDC2000T47）进行，该语料是来自香港的一些法律文本。我们使用了其中的 20 万平行句对来作为训练语料，并从这 20 万平行句对之外的部分随机选取了不重叠的 1000 个和 1558 个平行句对分别做开发集和测试集，并且考虑到模拟交互环境对参考译文准确性的要求，开发集和测试集的平行句对都是经过人工校正

的。表 1 示出了所用语料的一些统计特性。

中文部分都采用 ICTCLARS 进行了分词处理，并且所用语料的英文部分都经过了词形还原和小写化处理。GIZA++[8]工具被用来进行训练语料的词对齐工作，而双向词对齐的融合采用 Grow-Diag-Final 策略。此外我们利用 SRLIM[9]工具在训练语料的英文单语语料上训练了一个 3-gram 的语言模型。我们使用开源工具 moses 来训练基于短语的统计翻译模型。该短语模型使用了 moses 默认的 14 个特征，并且这些特征之间按照对数线性的方式进行结合，此外，我们使用了最小错误率训练[10]（MERT）来对特征的参数进行优化，并且优化指标采用大小写不敏感的 BLEU-4 指标。句法树采用 berkeley 句法分析器生成[11]，我们选用 1-best 句法树来抽取子树信息。

表 1 语料统计特性

		中文	英语
训练集	句子	200K	
	单词	5.15M	5.11M
	词汇	30K	31K
开发集	句子	1000	
	单词	15K	15.7K
	困惑度	73.24	48.65
测试集	句子	1558	
	单词	20.6K	21.6K
	困惑度	72.67	46.75

4.2 评价标准

在本文中，对交互式翻译系统的性能评价我们采用了 Key-stroke ratio（KSR）指标，该指标的计算方法为：用要得到标准译文（参考译文）所需的键盘敲击次数除以标准译文（参考译文）所包含的字符总数[7]。明显的是其数值越小，则交互式翻译系统的性能也应该越好。

4.3 系统设置

基线系统（Baseline）是我们实现的传统的交互式机器翻译系统，然后我们子树信息通过三种策略加入到基线系统中。三种策略分别表示为（+ISIBCP：在翻译假设未覆盖译文前缀时使用子树信息，+ISIACP：在翻译假设已经覆盖译文前缀后使用子树信息，+Both ：在整个翻译过程中使用子树信息）。为了更好的显示系统性能，我们在不同的 N-best 列表上计算评价标准。

4.4 实验结果及分析

表 2 不同系统的实验结果

系统	1-best KSR	5-best KSR	10-best KSR	20-best KSR
Baseline	48.66	47.93	47.76	47.55
+ISIBCP	48.05	47.41	47.18	46.97
+ISIACP	48.44	47.75	47.48	47.21
+Both	47.85	47.16	46.88	46.69

表 2 使用的系统中，对于上一节我们所提到的三个影响因素设置是一致的。这里我们在使用子树的选择上使用 min_subtree，对于另外的两个可能影响系统的性能没有考虑。通过表 2 的实验结果，我们可以看到，把子树信息嵌入到交互式翻译系统，无论是在覆盖翻译假设前，还是覆盖翻译假设后，都可以在一定程度上减少翻译人员的工作量，但在“+Both”系统中表现出比其他系统更好的性能。

表 3 不同系统的速度

系统	速度（句/秒）
Baseline	3.43
+ISIBCP	3.07
+ISIACP	3.23
+Both	2.86

表 3 是各个系统的在用 1-best 结果作为参考的情况下，翻译速度方面的表现，我们发现随着系统性能提高，在可接受的范围内，速度也会有所下降。

在其他影响因数固定的情况下，使用不同的策略选择子树。通过表 4 中的实验结果，我们知道，当子树出现嵌套的情况时，选取

不同的子树来评价当前的翻译假设，会对翻译假设的选取有一定的影响，同时对翻译系统的性能产生一定的影响。

表 4 max_subtree 和 min_subtree 的实验结果

系统	min_subtree KSR&Speed	max_subtree KSR &Speed
+Both	47.85/2.86	47.78/2.90

表 5 RatioSubtreeSentece 的实验结果

KSR	RatioSubtreeSentece				
系统	0.2	0.4	0.6	0.8	1.0
+Both	48.08	47.80	47.96	47.69	47.85

表5的结果是在其他影响因素固定的情况，根据 RatioSubtreeSentence 在抽取子树时，对当前句子中所包含的子树进行过滤，在遇到子树嵌套的情况使用 min_subtree 策略选择子树。并在“+Both”系统上进行实验，结果表明，在一定的情况下对子树进行过滤会提高系统性能。

表 6 的结果是我们在“+Both”系统基础上，对所选择符合子树限制的翻译假设的分值与当前的翻译假设的分值比做了限制，我们可以看到随着分值比的限制系统性能越来越差，这也从另一个方面显示了，调序模型的简单，未给出适当的分值。

表 6 score_ratio 的实验结果

KSR	score_ratio				
系统	0	0.2	0.4	0.6	0.8
+Both	47.85	47.88	48.05	48.16	48.22

5 总结及未来工作

本文我们针对统计机器翻译中的简单的调序模型，在交互式机器翻译的框架下，提出利用句法信息，子树信息，作为限制，选择更加合理的翻译假设进行扩展，且利用交互式机器翻译中特有的特征用户确定的译文前缀，提出三种不同的策略把子树信息加入到交互式翻译系统中。另外，我们还发现了几个影响系统性能的因素，如当子树出现嵌套时，子树的选择；抽取子树时子树包含词的个数与当前句子之间的比值；所选择的符合子树限制到翻译假设与传统的方法所选择的翻译假设之前的分值比。经过试验证明这些都会影响到系统的性能，上一节的实验中我们只验证了这些因素单独的使用对系统性能的可能影响。所以未来的工作会研究一下上一节提到的三种因素结合到一起对系统系能的影响。而且，在此系统中，对于由于各种原因不能匹配用户前缀的情况，系统会直接跳出解码，不会给出翻译后缀。因此，后面的研究也会涉及到在当前系统不能匹配用户给出翻译前缀时生成翻译后缀的策略。当前我们的基线系统采用的是多栈的解码策略，我们下一步将研究在柱搜索解码策略中子树信息的应用。

参考文献

[1] Foster G, Isabelle P, Plamondon P. Target-text Mediated Interactive Machine Translation[J]. Machine Translation, 1997, 12(1): 175-194

[2] Langlais P, Foster G, and Lapalme G. TransType: a Computer-aided Translation Typing System[C]. Proceedings of the NAACL/ANLP Workshop on Embedded Machine Translation Systems, 2000: 46-52

[3] Ortiz-Martinez D, Garcia-Varea I, Casacuberta F. Online Learning for Interactive Statistical Machine Translation[C]. Proceedings of NAACL 2010, 2010: 546-554

[4] Gonzalez-Rubio J, Ortiz-Martinez D, Casacuberta F. Balancing User Effort and Translation Error in Interactive Machine Translation Via Confidence Measures[C]. Proceedings of the 48th ACL, 2010: 173-177

[5] Gonzalez-Rubio J, Ortiz-Martinez D, Casacuberta F. Active learning for interactive machine translation[C]. Proceedings of the 13th EACL, 2012: 245-254

[6] Jesús González-Rubio, Daniel Ortiz-Martínez, José-Miguel Benedí, et al. Interactive Machine Translation using Hierarchical Translation Models[C].Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 244–254

[7] Och FJ, Zens R, and Ney H. Efficient Search for Interactive Statistical Machine Translation[C]. Proceedings of EACL 2003, 2003: 287-293

[8] Och, F. J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19-51

[9] Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at Sixteen: Update and Outlook. In Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop.

[10] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In Proc. of ACL,pages 160-167

[11] Petrov S, Barrett L, Thibaux R, et al. Learning accurate, compact, and interpretable tree annotation Association for Computational Linguistics, 2006: 433-440.