模糊匹配在树到串翻译模型中的应用*

熊皓　刘洋　刘群
中国科学院计算技术研究所 北京 100190
E-mail: {xionghao, yliu, liuqun} @ ict.ac.cn

摘要： 在传统的基于树的翻译模型中，一般都是将一条规则视为字符串，然后使用字符串匹配技术从规则表中搜索可用的规则。然而，由于基于树的翻译模型依赖于句法分析的结果，而有些语言的句法分析准确率并不是很高，所以句法分析错误造成的规则无法匹配的现象很常见，特别是在树到树的翻译模型中，能够精确匹配的规则数量非常稀少，进而对机器翻译的性能造成很大影响。因此本文提出了一种基于树核的模糊匹配技术，在 NIST 2005 汉英翻译测试集上的结果表明，基于树核的模糊匹配模型相对于传统的翻译模型显著的提高了 1.3 个 BLEU 值，并且在森林模型中使用模糊匹配技术仍然能够提高 0.7 个 BLEU 值。

关键词： 树核，树到串翻译模型，统计机器翻译，模糊匹配

Fuzzy Matching for Tree-based Machine Translation

Xiong Hao　Liu Yang　Liu Qun
Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190
E-mail: {xionghao, yliu, liuqun} @ ict.ac.cn

Abstract: Previous related work of tree-based models treat rules as strings and then match rules using string matching algorithm. However, the performance of tree-based models is largely depended on the parsing results, and for some languages, the precision of current parser is still far from state-of-the-art. So two rules with one different tag causing by parsing errors seems to be unmatchable. Under exact matching strategy, the size of available rules is implicitly scarce especially in tree-to-tree models, in which the performance is still unacceptable. In this paper, we present a tree kernel based fuzzy matching algorithm which computes the similarity between different rules. Experimental results on NIST 2005 Chinese-to-English test set show that our system achieve an absolute improvement of 1.3% in term of BLEU score over string matching system. Furthermore, when using the packed forest, our method still gets a relative improvement of 0.7 BLEU score.

Keywords: tree kernel, tree-to-string model, statistical machine translation, fuzzy matching

1 导论

最近几年来，基于树的翻译模型受到了越来越多的关注，并且在近几年的NIST翻译评测中取得了不错的成绩。根据输入的不同，基于树的模型可以分为以下两类：串输入模型 (Chiang, 2005; Wu, 1997; Galley et al., 2006; Marcu et al., 2006) 和树输入模型 (Lin, 2004; Ding and Palmer, 2005)。串输入模型使用上下文同步文法对输入的文本串同时进行句法分析和翻译；而树输入模

* 本文受国家自然科学基金重点项目《融合语言知识与统计模型的机器翻译方法研究》（批准号 60736014）以及 863 重点项目《面向跨语言搜索的机器翻译关键技术研究》（课题编号 2006AA010108）资助。

型则直接将输入的句法树转换成目标翻译或者目标句法树。

树输入的模型主要包括树到串翻译模型 (Liu et al., 2006; Huang, 2006) 和树到树翻译模型 (Eisner, 2003; Cowan, 2006;Zhang et al., 2008)。这两种模型都将解码部分分为两个步骤：句法分析和翻译。首先使用句法分析器将输入的源文本串分析成一棵句法分析树，然后利用解码器将句法树转换成目标翻译。然而对于某些语言来说，比如中文，句法分析的准确率远没有达到令人满意的结果。因此，解码器的性能不可避免的要受句法分析错误的影响 (Quirk and Corston-Oliver, 2006)。

为了减轻句法分析错误对翻译性能的影响，一种可行的方法是在源端使用句法分析森林来替代单棵最优句法树 (Mi et al., 2008; Zhang et al., 2009; Liu et al., 2009)。上述文献也表明使用森林技术可以很大程度上提高翻译的质量。但是不管是使用单棵句法分析树还是使用森林，以往的工作都是将规则表示为文本串，然后使用字符串匹配技术从规则表中搜索可用的规则。但是有些规则所含的节点数目非常多，如果使用字符串匹配技术进行完全匹配，从规则表中很难找到可用的规则。例如，对于如下两条规则：

(NP-B (NR (中国) **NN** (打击) NN (走私) NN (工作) NN (会议))

(NP-B (NR (中国) **VV** (打击) NN (走私) NN (工作) NN (会议))

由于句法分析的错误，第二条规则中“打击”的词性被标注成了“VV”，按照字符串匹配算法，这两条规则是无法匹配的。从上面这个例子可以看出，使用完全的字符串匹配技术将潜在地减小可匹配的规则数量，特别对于一些节点数目比较大的规则来说，能够成功匹配的规则数量将非常稀少。

因此本文提出了一种基于树核的模糊匹配技术，我们将一条规则表示成一个特征向量，通过卷积树核 (Collins and Duffy, 2001) 来计算不同规则之间的相似度。由于规则表中规则数量巨大，完全计算所有规则之间的相似度是不可行的，因此我们首先通过一些限制生成一个候选规则集合，然后在集合内部通过卷积树核计算相似度。在NIST 2005汉英翻译的实验表明，使用模糊匹配技术显著的提高了1.3个BLEU值，并且在森林模型中使用模糊匹配技术仍然取得了0.7个BLEU值的提升。

本文将在第二节简要介绍树到串翻译模型，然后在第三节介绍卷积树核以及本文提出的模糊匹配技术，在第四节本文将给出所有的实验结果，最后在第五节将对本文进行总结。

2 树到串翻译模型

树到串翻译模型 (Liu et al., 2006; Huang,2006) 将翻译过程分成两个步骤：句法分析和翻译。假定源语言是中文，目标语言是英文，图1给出了一个树到串翻译模型中常见的例子，其中包括源句法分析树，目标串以及源端和目标端文本串之间的对齐信息。

使用GHKM算法 (Galley et al.,2004)，我们可以从图1中的对齐结构中抽取出树到串翻译规则。表1给出了图1例子中抽取出的规则样例。其中规则的左半部分表示的是一棵树片段（x表示的是非终结符）；右半部分是对应的翻译信息。由于树到串规则包含了丰富的层次信息，因此其表达能力也强于同步上下文文法。

通过在训练集中抽取出所有的翻译规则可以形成一个规则表，然后解码器在句法分析树的所有推导 D 中搜索最优推导 d^*，通过匹配规则表中的翻译规则，将推导转换成目标翻译。因此，

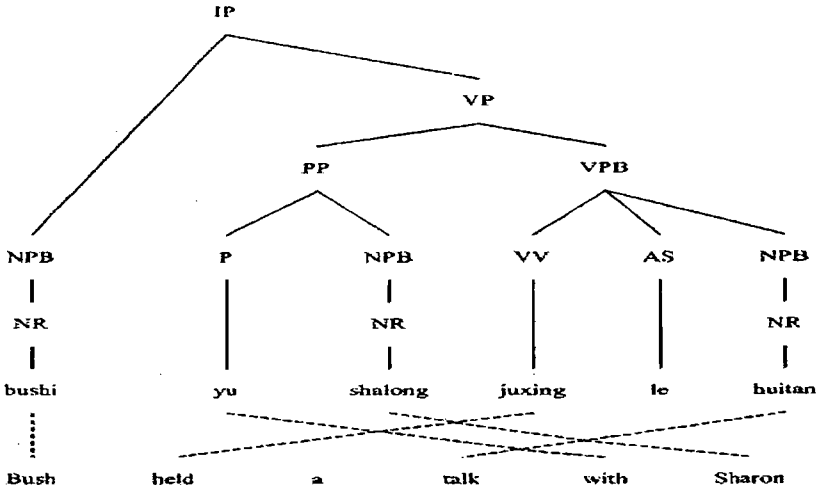

图1： 树到串翻译模型中的常见例子，其中包括源句法分析树，目标串以及源端和目标端之间的对齐信息

(1) IP(x_1:NPB VPB(x_2:PP x_3VPB)) → $x_1x_3x_2$
(2) NPB(NR(*bushi*)) → *Bush*
(3) PP(P(*yu*) x_1:NPB) →*with* x_1
(4) NPB(NR(*shalong*)) → *Sharon*
(5) VPB(VV(*juxing*) AS(*le*) x_1:NPB) → *held a* x_1
(6) NPB(NN(*huitan*)) → *meeting*
(7) NR(*shalong*) → *Sharon*

表1：规则表

树到串翻译模型的目标是优化目标函数：$d^* = \underset{d \in D}{\operatorname{argmax}} P(d \mid T) = \underset{d \in D}{\operatorname{argmax}} \prod_{r \in d} P(r)$ ，其中 r 是规则表中匹配的规则。

在以往相关的工作中，表1中的规则(4)和规则(7)是两条完全不同的规则，因为他们的左半部分不能完全匹配。但是我们可以发现他们表达的翻译信息是相同的，都是将源端的词汇化节点“*shalong*”翻译成目标端的“*Sharon*”。因此我们提出了一种模糊匹配技术，我们只考察规则中的边缘节点（规则中的叶子节点）相同的规则，然后使用树核计算规则之间的相似度并作为一个判别特征加入到判别模型中去。在下面一节中我们将详细介绍基于树核的模糊匹配技术。

3 模糊匹配

在我们的系统中，一条规则不再表示成字符串，而是将其视为一个结构体，然后计算不同规则之间的相似度。为了计算相似度我们引入了卷积树核技术。本节我们首先简要介绍卷积树核的基本概念，然后介绍如何将卷积树核技术引入到模糊匹配模型中。

3.1 卷积树核

不同于树结构的字符串表示形式，卷积树核 (Collins and Duffy, 2001) 通过特征向量来表示不同的句法树。一般来说，一棵句法树t可以使用特征向量f来表示，f具有如下形式：

$f(t)=(...,st_i(t),...)$，其中$st_i(t)$表示的是句法树t中第i^{th}棵子树出现的次数。由于一棵句法树中的子树个数有可能非常多，直接枚举是不可能的，因此Collins and Duffy (2001) 提出了使用卷积树核来高效计算高维向量点积的方法：

$$K(t_1,t_2)=\langle f(t_1),f(t_2)\rangle=\sum_i\sum_{n_1\in N_1}I_i(n_1)\cdot\sum_{n_2\in N_2}I_i(n_2)=\sum_{n_1\in N_1}\sum_{n_2\in N_2}C(n_1,n_2)$$

其中N_1和N_2分别是句法树$t1$和$t2$的节点集合，$I_i(n)$表示句法树的子树是否以n作为根节点，是则为1，反之为0；$C(n_1,n_2)$表示两棵句法树中分别以n_1和n_2作为根节点的子树个数。并且$C(n_1,n_2)$可以通过下面的定义在多项式时间内计算出来：

1）如果节点n_1和n_2的推导不同，则$C(n_1,n_2)=0$

2）如果节点n_1和n_2为叶子节点并且具有相同的标签，则$C(n_1,n_2)=\lambda$

3）如果节点n_1和n_2的推导相同并且n_1和n_2不是叶子节点，则 .

$$C(n_1,n_2)=\lambda\prod_{j=1}^{nc(n_1)}(1+C(ch(n_1,j),ch(n_2,j)))$$

其中$nc(n_1)$表示节点n_1包含的推导中子节点个数，由于节点n_1和n_2的推导相同，所以$nc(n_1)=nc(n_2)$，此外$ch(n_1,j)$表示n_1包含的推导中第j个子节点。λ（$0\le\lambda\le1$）是一个惩罚因子，用来降低子树规模对$C(n_1,n_2)$大小的影响。

3.2 规则匹配

在以往的方法中，规则必须通过精确匹配来选取，以图2为例：假定图中表示的是输入句法树中的两棵不同子树片段。通过规则匹配，我们发现表1中的规则(5)可以用来生成子树a的翻译，而子树b无法从表1的规则表中找到合适的翻译，因为表1包含的所有规则表中没有任何一条规则的左半部分包含子树片段 “VPB(VV(*jiuxing*) AS(*le*) x1:NP)”。

但是通过分析我们发现，子树a和子树b的翻译仅由叶子节点“juxing”，“le”，“NPB”和“NP”的翻译译文组成，而与中间节点“VV”和“AS”无关。基于这个直觉，我们忽略子树的中间结构，仅考察子树的叶子节点是否相似，并且对于非终结符节点进行泛化处理。例如图2中子树a和子树b的“NBP”和“NP”节点都泛化为名词词性“NNx”。同样地，对于规则表中的规则，我们匹配时也仅考虑规则中的叶子节点，以表1中的规则(5)为例，其对应的叶子节点为“juxing”，

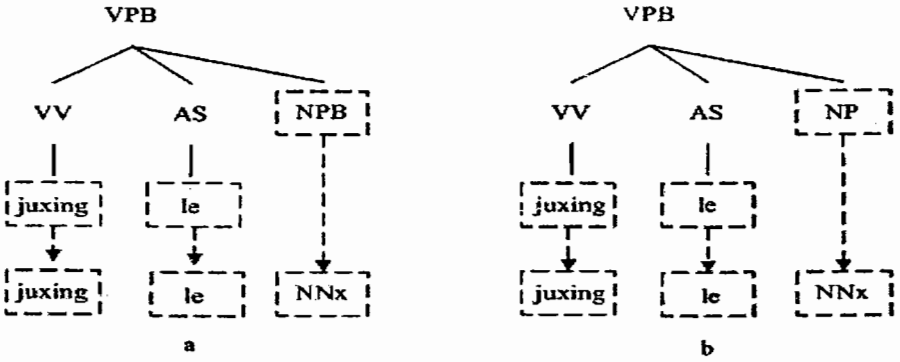

图2：树到串翻译模型中输入句法树中的两棵子树片段及其在模糊匹配模型中的表示形式

“le”和“NP”，由于“NP”为非终结符名词节点，我们将其泛化为“NNx”。因此规则5在模糊匹配的定义下适用于图2中的两棵子树。

然而对于规则表中的规则来说，由于我们只考察叶子节点是否相似，而忽略中间节点的不同。但是对于一条和源端子树完全匹配的规则来说，规则的可信度要高于不完全匹配的规则。因此对于不同的匹配规则，我们使用上一小节介绍的卷积树核来计算输入子树片段和规则表中规则之间的相似度，并在解码时将相似度作为一个特征引入到对数线性模型中去，通过最小错误率训练来训练出合适的权重。在实验中我们也发现模糊匹配的权重为正值，也就意味着越相似的规则对BLEU贡献越大。

此外由于计算树核相似度需要耗费一定的计算时间，而对于解码器来说性能是很重要的。因此为了提高解码时的运行速度，我们将规则表中叶子节点相同的规则形成一个小集合，并且利用树核公式预先计算出集合内规则之间的相似度，这样在解码时就可以快速获取子树和匹配规则之间的相似度，省去了在线计算树核的时间。

4 实验

4.1 实验设置

我们选用FBIS语料（共239416平行句对）作为训练集，利用SRI语言模型工具包(Stolcke, 2002a)，并且使用Kneser-Ney 平滑(Chen and Goodman, 1996)技术在GigaXinHua上面训练了一个4元语言模型。

对于源端的汉语句子，我们使用中文句法分析器(Xiong et al., 2005) 生成了对应的中文句法树。此外我们选用NIST2002 汉英测试集作为实验的开发集，NIST2005汉英测试集作为测试集，使用大小写不敏感的BLEU4作为实验衡量标准。

4.2 实验结果

由于使用了模糊匹配，对于源端的句法树来说，可候选的规则数量相比于精确匹配增加12%左右，表2列出了精确匹配和模糊匹配对应的候选规则数量。

我们在开发集中对3.1节中相似度公式中的惩罚因子λ进行插值计算，图3给出了开发集

	开发集（NIST02）	测试集（NIST05）
精确匹配	1,430,979	1,624,456
模糊匹配	1,608,347	1,816,598

表2：精确匹配和模糊匹配模型过滤后规则表大小

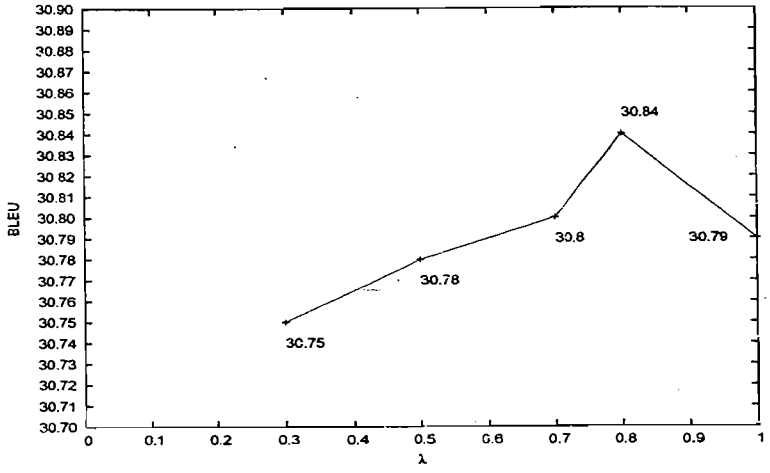

图3：不同的 λ 值对开发集BLEU值的影响

	测试集05 (BLEU)
树模型精确匹配（baseline）	28.76
树模型模糊匹配	**30.06**$^{++}$（$p<0.01$）
森林模型精确匹配（baseline）	30.7
森林模型模糊匹配	**31.4**$^{++}$（$p<0.05$）

表3：树模型和森林模型在测试集上的对比实验结果

BLEU值受 λ 值的影响，从图中可以看出当 λ =0.8时我们取得了最优的BLEU值，但是可以看出树核函数的计算方法对最后的性能影响并不是很大，主要原因是在对数线性模型中，规则的选取取决于多个特征的判别作用。相对于翻译概率以及词汇化概率来说，相似度特征的作用并不是很大。但是从表3表4中我们发现仅考虑叶子节点的匹配策略对于BLEU的提升很有效。

最后我们测试了模糊匹配在测试集上面的效果，表3给出了对比实验结果。从表3中我们可以发现模糊匹配相对于精确匹配取得了1.3个BLEU值的显著提升。提升的很大原因在于树到串翻译模型的源端是句法树，而在目前句法分析准确率不高的情况下，传统的精确匹配方法无法减轻句法分析错误对规则无法匹配造成的影响。

此外，我们在基于森林的树到串模型 (Mi et al., 2008) 中引入了模糊匹配技术，从表3中可以看出基于模糊匹配的技术仍然取得了0.7个BLEU值的提升。从表3对比结果中我们可以发现使用模糊匹配技术能够显著的提高机器翻译的性能。

5 结论与未来工作

本文提出了一种基于树核的规则模糊匹配技术，在NIST05的测试集合上的实验结果表明：相比于传统的精确匹配方法，使用模糊匹配技术显著的提高了1.3个BLEU值；此外，在基于森林的翻译模型上使用模糊匹配技术，我们仍然取得了0.7个BLEU值的提升。

在未来的工作中我们将继续测试模糊匹配技术在树到树翻译模型中的效果，相比于树到串模型，树到树模型目前的翻译效果仍然有很大潜力需要挖掘，因此我们希望通过使用模糊匹配技术来增加可用规则表大小，进而提高翻译的质量。

参 考 文 献

Stanley Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. *ACL*, pages 310–318.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. *ACL*, pages 263–270.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. *NIPS*, pages 625-632.

Brooke Cowan. 2006. A discriminative model for tree-to-tree translation. *EMNLP*, pages 232-241.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. *ACL*, pages 541-548.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. *ACL*, pages 205-208.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What is in a translation rule? *NAACL*, pages 273-280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. *ACL*, pages 968-976.

Liang Huang. 2006. Statistical syntax-directed translation with extended domain of locality. *AMTA*, pages 66–73.

Dekang Lin. 2004. A path-based transfer model for machine translation. *COLING*, pages 625-633.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. *ACL*, pages 609-617.

Yang Liu, Yajuan L¨u, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. *ACL*, pages 558-566.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. Spmt: Statistical machine translation with syntactified target language phrases. *EMNLP*, pages 44-52.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. *ACL*, pages 206-214.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. *Proceedings of International Conference on Spoken Language Processing*, pages 901–904.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the penn Chinese treebank with semantic knowledge. *IJCNLP*, pages 70–81.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. *ACL-08*, pages 559-567.

Hui Zhang, Min Zhang, Haizhou Li, Aiti Aw, and Chew Lim Tan. 2009. Forest-based tree sequence to string translation model. *ACL*, pages 172-180.