半结构化的基于实例的机器翻译

王海峰　刘占一　吴华

东芝(中国)研究开发中心 北京 100738

E-mail: {wanghaifeng, liuzhanyi, wuhua}@rdc.toshiba.com.cn

摘　要：本文介绍了一种半结构化的基于实例的机器翻译方法。该方法将双语翻译实例表示为半结构化的树串映射，一个树串映射由三部分组成：源语言分析树、目标语词串、及它们之间互译词的对应关系。基于这种树串映射进行机器翻译时，首先使用源语言分析器得到待翻译句子的分析树；然后在翻译实例库中搜索与该分析树最匹配的树串映射森林；最后，使用线性对数生成模型产生译文。使用 BLEU 作为机器翻译评价准则的实验结果表明，本文方法显著优于著名的基于短语的统计翻译系统 Pharaoh。

关键词：树串映射，翻译实例，基于实例的翻译，机器翻译

Semi-Structured Example Based Machine Translation

WANG Haifeng, LIU Zhanyi, and WU Hua

Toshiba (China) Research and Development Center, Beijing, 100738

E-mail: {wanghaifeng, liuzhanyi, wuhua}@rdc.toshiba.com.cn

Abstract: This paper describes a semi-structured example based machine translation (EBMT) method. In this method, the translation example is represented as a Tree String Correspondence (TSC), which is a triple consisting of a parse tree in the source language, a string in the target language, and the correspondence between the leaf node of the source-language tree and the substring of the target language string. An input sentence is first parsed into a tree. Then the TSC forest which best matches the input tree is searched for. Finally the translation is generated using a log-linear generation model to combine the target-language strings of the TSCs. Using BLEU as a metric, our method is significantly better than Pharaoh, which is a famous phrase-based statistical machine translation system.

Keyword: tress string correspondence, translation example, example based machine translation, machine translation

1　前言

基于实例的机器翻译（Example Based Machine Translation，简称 EBMT）的本质是“以翻译实例为基础，基于相似原理的机器翻译”，其利用的主要知识源是预处理过的双语语料和翻译词典[1]。典型的 EBMT 模型通常包括三个模块：在翻译实例库中搜索匹配片段；确定相应的译文片段；重新组合译文片段以得到最终翻译[2]。

本文介绍了一种半结构化的基于实例的机器翻译方法。该方法将双语翻译实例表示为半结构化的树串映射（Tree-String-Correspondence，简称 TSC），一个 TSC 由三部分组成：源语言分析树、目标语词串、及它们之间互译词的对应关系。基于 TSC 进行机器翻译时，首先使用源语言分析器得到待翻译句子的分析树；然后在翻译实例库中搜索与该分析树最匹配的 TSC 森林；最后，使用线性对数生成模型产生译文。

近些年来，EBMT 系统普遍采用结构化的双语例句来表示翻译实例[3][4][5][6][7]。对于这样的翻译实例，翻译系统在双语例句预处理阶段必须同时分析源语言句子和目标语句子得到结构信息，而且要对齐源语言和目标语分析树。结构化的翻译实例包含了丰富的句法信息，有利于翻译质量的提高。但是，建立这样一个双语结构化的翻译实例库是非常困难的。首先，对于很多语言来讲，不存在正确率较高的分析器；其次，在两个不同语系的语言之间，例如英语和汉语，很难建立句子结构之间的完全对应关系[5][8]。本文使用半结构化的 TSC 来表示和存储翻译实例。相

比于双语完全结构化的方法，该方法只使用源语言分析器，并且只需对齐源语言和目标语言的词。

在使用结构化翻译实例的 EBMT 系统中，通常采用编辑距离来计算输入句子和翻译实例之间的相似度[9][10][11]。这种计算需要引入非常复杂的树匹配操作，从而导致很低的翻译效率。本文使用贪心算法来搜索与源语言分析树最佳匹配的 TSC 森林，使匹配效率显著提高。

通常有两类EBMT译文生成算法：基于语义和基于统计的方法。前者通过语义词典获得每个源语言片段对应的译文，然后按照预先定义好的顺序来合并这些译文片段生成最终译文[12][13]。这种方法没有考虑译文片段之间的衔接，因此译文流利度会受到影响。基于统计的生成方法利用统计模型来选择/合并译文片段生成最终译文[14][15][16][17] [18] [19]。这种方法一定程度上解决了译文片段的衔接问题，但没有考虑译文片段和输入句子的语义关系，从而影响了译文生成的正确性。本文提出了一个线性对数（log-linear）的 EBMT 生成模型，该模型使用了六个特征：用来估计输入句子和翻译实例之间相似程度的匹配得分和上下文相似度；用来估计翻译实例可靠性的词翻译概率和目标语词串选择概率；用来估计生成译文质量的语言模型概率和长度选择概率。

使用 BLEU 作为机器翻译评价准则的实验结果表明，本文方法显著优于著名的基于短语的统计翻译系统 Pharaoh。

2 树串映射

首先，我们定义匹配树。对于一个分析树 T 和其中的子树 T_s 来说，如果 T_s 是 T 的一个匹配树，那么 T_s 必须满足以下三个条件：

- T_s 至少包含两个节点；
- T_s 中只存在一个根节点 r，其父节点不在 T_s 中，且 T_s 中其它节点都是 r 的后继节点；
- 对于 T_s 中除 r 外的任意节点，其兄弟节点也必须在 T_s 中。

在本文中，分析树中每个节点包含两部分：中心词和词性。

TSC 由一个三元组<*t*, *s*, *c*>组成，其中，*t* 是源语言匹配树，*s* 是目标语词串，*c* 是 *t* 中的叶子节点和 *s* 中的子串之间的对应关系。

如果 TSC 中的匹配树的叶子节点是其分析树的一个非终节点，那么这种叶子节点叫做替换节点，其在目标语词串中对应的部分叫做替换符号。在翻译过程中，目标语中的替换符号会被其它 TSC 的译文所替换。

TSC 可以用来表示静态翻译实例或动态双语翻译片段。在 EBMT 系统中，预处理过的双语例句被存储成 TSC；在翻译过程中，和输入句子匹配的双语翻译片段则用 TSC 来表示。

图 1 显示了三个英语到汉语的 TSC。TSC (a) 表示了下面的双语例句：

英语句子：Mary borrowed a book from her friend.
汉语句子：玛丽 从 她 朋友 那里 借 了 一 本 书 。
词对齐： (1:1) (2:6) (3:8) (4:10) (5:2) (6:3) (7:4) (8:11)

在图 1(a)中，源语言匹配树和目标语词串分别由例句的源语言部分和目标语部分组成。而(b)和(c)则是从双语例句中导出的两个 TSC。它们的源语言匹配树是(a)中根节点为 4 的子树。(b)中匹配树包含了其根节点的所有后继节点，匹配树中不包含替换节点，对于这样的 TSC，目标语词串认为是匹配树的译文。而(c)中的匹配树，节点 6 和 11 在(a)的分析树上是非终节点，所以在(c)中，这两个节点是替换节点，它们在目标语词串中对应的“<NPB>”是替换符号。对于像(c)这样的 TSC，目标语词串由单词和替换符号组成，没有直接对应的译文片段，其译文只能通过使用其它 TSC 的译文来替换所有替换节点而得到。

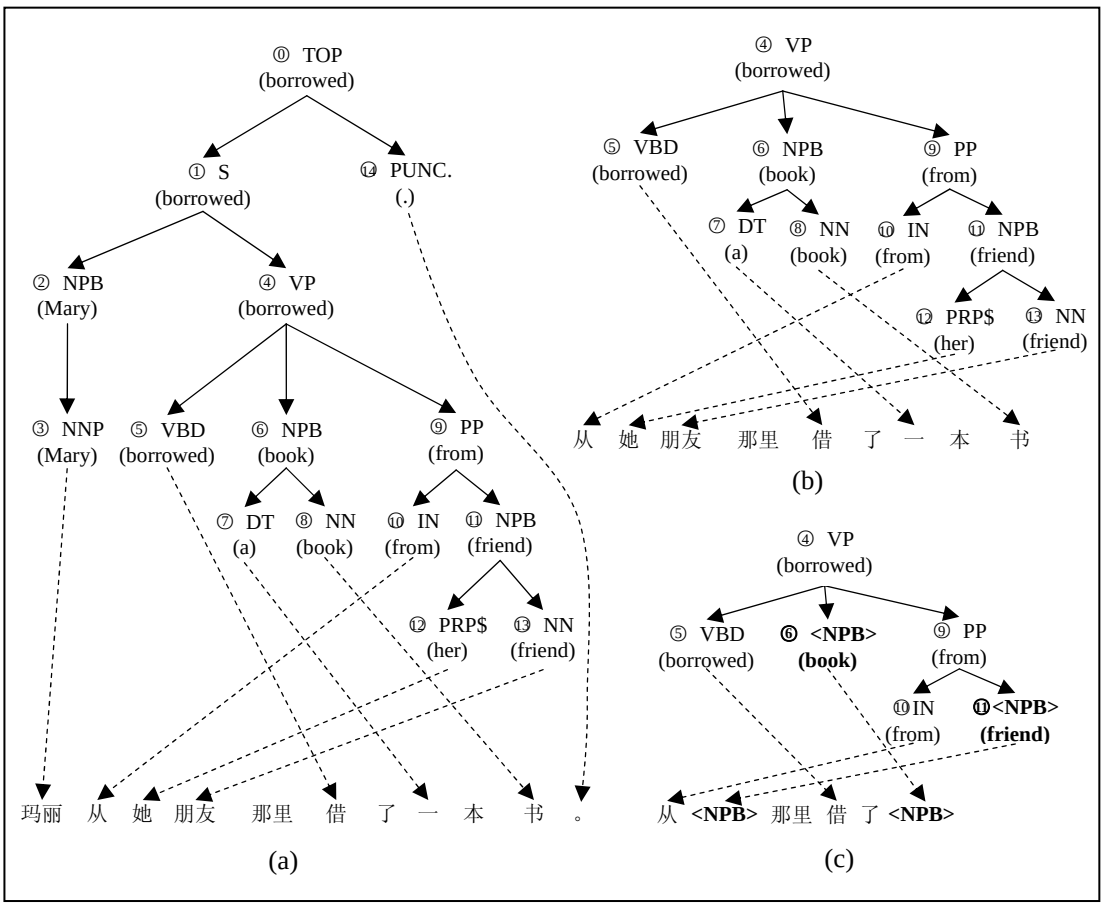

图 1. 树串映射的例子

如果两个 TSC 的源语言匹配树完全相同，而目标语词串不同，那么称这两个 TSC 是同源的。利用同源的 TSC，我们可以得到不同目标语词串，进而得到不同译文。

一个 TSC 森林能匹配一个分析树意味着 TSC 森林中的源语言匹配树能严格地组成该分析树。对于 TSC 森林里的两个 TSC T_1 和 T_2 来说，如果 T_1 的根节点匹配 T_2 的一个替换节点，那么 T_1 称为 T_2 的子 TSC；T_2 称为 T_1 的父 TSC。

3 TSC-Tree 匹配算法

对于给定的分析树 *T* 来说，一个<*t*, *s*, *c*>和该分析树是匹配的，则需满足下列条件：

1) *t* 中的非终节点和其在 *T* 中对应节点的中心词和词性必须一致；
2) *t* 中的叶子节点和其在 *T* 中对应节点的词性必须一致；

3.1 匹配得分

在翻译实例库中，可能不止一个 TSC 与给定的分析树匹配，我们通过计算 TSC 的匹配得分来选择匹配整棵分析树的 TSC 森林。对于分析树和与之匹配的 TSC，本文有如下的假设：

1）TSC 包含的节点个数越多，则 TSC 质量越好；
2）TSC 中替换节点和分析树中对应节点之间的语义相似度越近，则 TSC 质量越好。

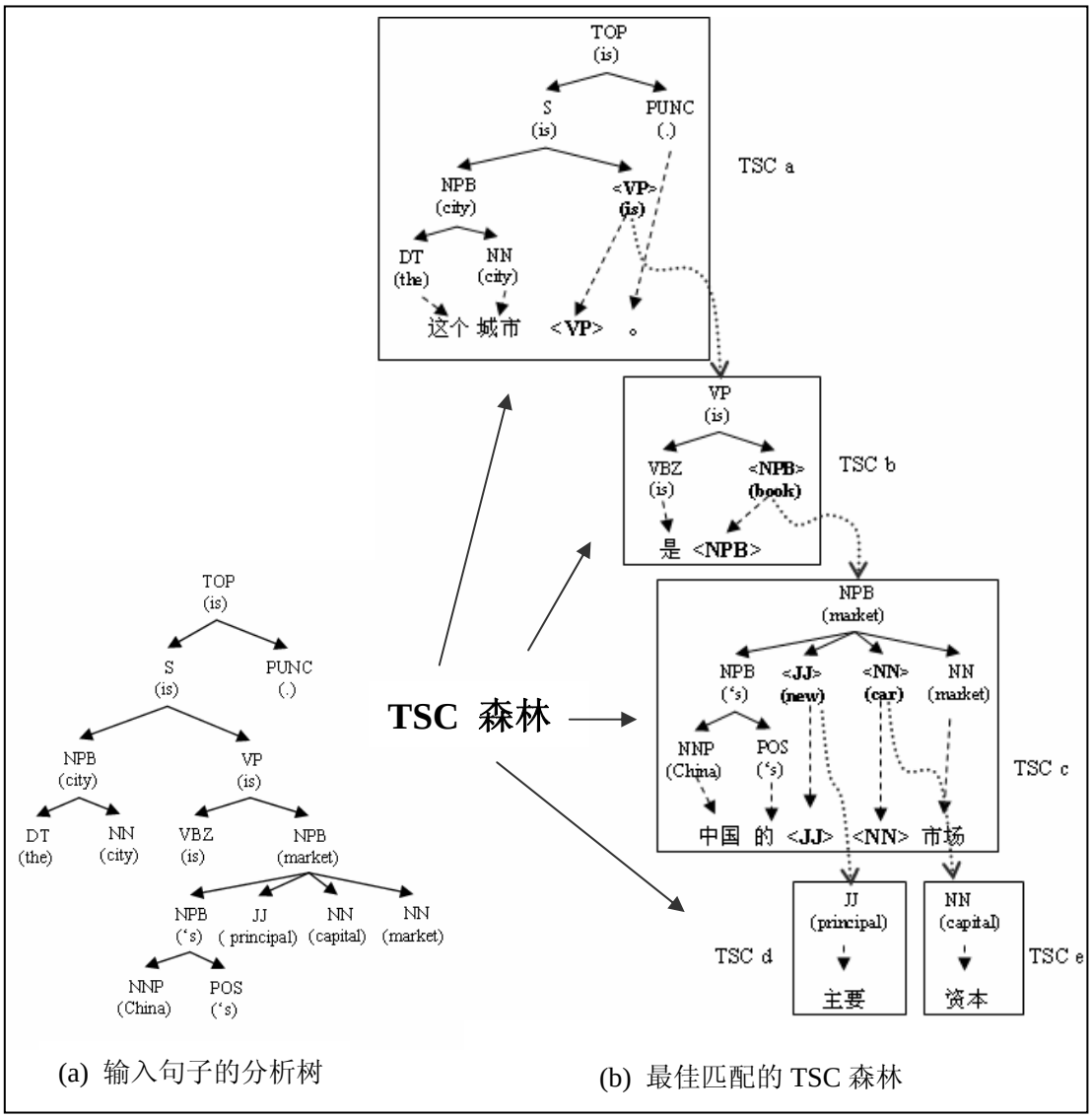

图 2. TSC-Tree 匹配算法示例

基于此假设，TSC 和分析树之间的匹配得分定义为它们之间对应节点的相似度总和，如果两个节点的中心词和词性都相同，那么节点的相似度等于 1，否则，相似度等于两个节点的中心词之间的语义相似度，本文采用了语义层次网络 WordNet 来计算英语单词之间的语义距离[20]，计算方法与文献[21]相同。

3.2 匹配算法

如果一个 TSC 森林中源语言匹配树恰好能组成给定分析树，那么我们称该 TSC 森林能匹配该分析树。对于一个分析树来说，翻译实例库中可能存在不止一个 TSC 森林能匹配该分析树，我们必须找到一个 TSC 森林，使它最佳地匹配给定的分析树。我们假设在所有候选 TSC 森林中，具有最高平均匹配得分的 TSC 森林是和给定的分析树最佳匹配的。

在本文中，我们采用了一种贪心的 TSC-Tree 匹配算法，来寻找和分析树最佳匹配的 TSC

森林。尽管贪心算法的结果不总是最好的，但是贪心算法能平衡翻译质量和翻译效率。

匹配算法采用自上而下策略来寻找匹配的 TSC。给定分析树，匹配算法首先在和分析树的根节点匹配的候选 TSC 中，找到一个具有最高匹配得分的 TSC，然后，对于该 TSC 中的每个替换节点，在翻译实例库中寻找匹配得分最高的子 TSC。反复执行这个过程，直到所有的替换节点都被扩展找到对应的子 TSC。

图 2 举例说明了怎样利用 TSC-Tree 匹配算法来寻找最佳匹配的 TSC 森林。输入的句子是：

The city is China's principal capital market.

图 2(a)表示输入句子的分析树，图 2(b)表示匹配的 TSC 森林。在图 2(b)中，搜索算法首先寻找能匹配分析树根节点的 TSC，在所有候选 TSC 中，TSC a 具有最高的匹配得分，所以 TSC a 被选中加入到 TSC 森林中。在 TSC a 中，有一个替换节点，对应分析树中的节点“VP(is)”，匹配算法继续搜索能匹配“VP(is)”的 TSC，TSC b 在所有的候选中具有最高的匹配得分，所以 TSC b 被加入到 TSC 森林中。TSC b 中也含有替换节点，类似地，搜索算法继续扩展替换节点，直到所有替换节点都被扩展为止。

在图 2(b)中，TSC b 的根节点对应 TSC a 的一个替换节点，所以 TSC a 是 TSC b 的父 TSC，TSC b 是 TSC a 的子 TSC。

4 译文生成

4.1 线性对数（Log-linear）生成模型

对于给定的输入句子 $\mathbf{f}=f_1^J=f_1,...,f_j,...,f_J$，具有最高概率的译文 $\mathbf{e}=e_1^I=e_1,...,e_i,...,e_I$ 可以通过下面的公式得到：

$$\mathbf{e} = \arg\max_{\mathbf{e}'}\{p(\mathbf{e}'\,|\,\mathbf{f})\} \tag{1}$$

本文采用最大墒方法直接为 $p(\mathbf{e}\,|\,\mathbf{f})$ 建模[22]。假设在模型中有M个特征函数$h_m(\mathbf{e},\mathbf{f})$, $m=1,...,M$，每个特征函数对应一个参数 λ_m。 $p(\mathbf{e}\,|\,\mathbf{f})$ 可以描述为：

$$p(\mathbf{e}\,|\,\mathbf{f}) = \frac{\exp[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e},\mathbf{f})]}{\sum_{\mathbf{e}'}\exp[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f})]} \tag{2}$$

这样，我们可以得到下面的判别规则：

$$\mathbf{e} = \arg\max_{\mathbf{e}'}\{\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f})\} \tag{3}$$

在公式(3)中，我们引入了下面的隐含变量来扩展特征函数：输入句子的分析树 **F** 和具有 *K* 个 TSC 的森林 **Z**。这样，我们就得到 *M* 个形式如 $h_m(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z})$的特征函数和下面的规则：

$$\mathbf{e} = \arg\max_{\mathbf{e}',\mathbf{F},\mathbf{Z}}\{\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f},\mathbf{F},\mathbf{Z})\} \tag{4}$$

在本文介绍的翻译系统中，对于一个待翻译的句子，只有一棵分析树被考虑。因此，公式(4)可简化为：

$$\mathbf{e} = \arg\max_{\mathbf{e}',\mathbf{Z}} \{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}', \mathbf{f}, \mathbf{F}, \mathbf{Z}) \} \quad (5)$$

4.2 特征函数

在线性对数生成模型中，我们使用了六个特征函数。这六个特征函数分别描述输入句子和翻译实例之间的相似度、翻译实例的可靠性、和生成译文的质量。

匹配得分和上下文相似度

这两个特征函数用来估计输入句子和翻译实例之间的相似度。TSC 和分析树之间的匹配得分用来描述匹配部分的语义距离，这部分已经在 3.1 节中介绍，这里不再重复。输入句子和翻译实例之间的上下文相似度主要用于选择合适的翻译实例来提高译文选择能力，它被定义成句子之间基于词的余弦距离。

词翻译概率和目标语词串选择概率

这两个特征函数估计翻译实例的可靠性。源语言和目标语之间的词翻译概率描述了 TSC 中词对应关系的质量，它从词对齐的双语语料中训练得到。目标语词串选择概率用于衡量目标语词串的可靠程度。给定源语言匹配树，我们用目标语词串的出现频度来计算目标语词串选择概率。

语言模型概率和长度选择概率

这两个特征函数用来估计生成译文的质量。语言模型概率刻画了生成译文的流利度，我们采用了目标语的三元语言模型来计算生成译文的概率，并使用概率的几何平均数来防止语言模型概率选择较短句子的倾向。我们引入长度选择概率来避免选择过短或过长的译文。长度选择概率描述了给定源语言句子长度的情况下目标语句子长度的概率。在翻译的过程中，除了估计译文长度，我们还需要估计译文片段长度。和句子相比，片段的长度更加灵活，也更难准确估计。对于给定的源语言长度，目标语的长度符合正态分布[23]。我们使用正态分布函数来近似地为译文片段的长度选择建模，我们利用双语例句得到正态分布的平均值 μ =1.03 和方差 σ^2 =5.56。

4.3 搜索

利用 TSC-Tree 匹配算法找到 TSC 森林后，我们利用线性对数生成模型来得到最终译文。为了引入更多候选译文，我们把同源的 TSC 添加到 TSC 森林中，从而扩展了 TSC 森林里的每个 TSC。

在扩展的 TSC 森林里，我们使用 Beam 搜索算法自底向上生成译文。图 3 显示了一个搜索过程。对于 TSC 森林中的每个 TSC，如果其目标语词串中包含替换符号，那么每个替换符号应该被对应的子 TSC 的候选译文替换，然后，通过线性对数生成模型的得分来对生成的候选译文排序，选择得分最高的前 *n* 个译文，这些译文再用来生成其父 TSC 的译文。

在所有与输入分析树的根节点匹配的候选译文中，得分最高的译文被作为输入句子的最终译文。

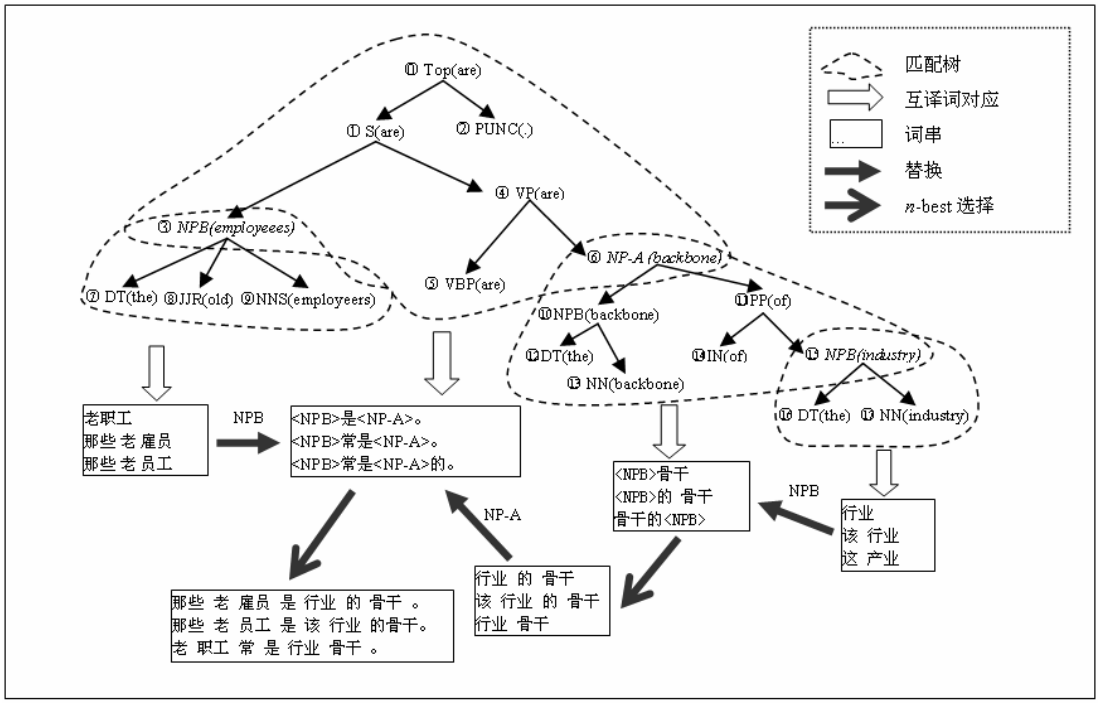

图 3. 搜索示例

5 实验和结果

为了验证基于 TSC 的 EBMT 模型的性能，我们开发了一个英语到汉语的翻译系统。我们采用 Powell 算法在开发集上训练特征函数的权重[24]，为了避免陷入局部最优，我们从不同的初始点来开始训练。在本文的实验中，搜索过程最多保留 100 个候选译文。

本文使用 BLEU 得分来评价译文质量[25]，同时，也为所有的翻译结果计算了 95%的置信区间[26]。

5.1 资源

双语例句库：例句库包含 262,600 个英汉句子对。英语句子的平均长度为 12.1 个单词；汉语句子的平均长度是 12.5 个单词（21.8 个汉字）。并使用 GIZA++工具进行词对齐。

测试集和开发集：测试集包含 400 个英语句子，开发集包含 100 个英语句子，它们都没有被包含于双语例句库中，每个句子有两个参考译文。

翻译词典：一个通用的英汉词典。该词典用来翻译输入句子中没有被双语语料库覆盖的单词。

语言模型：使用 SRILM 工具在通用汉语语料上训练的基于词的三元模型[27]。

英语分析器：采用了 Collins 的英语分析器[28]。

5.2 实验结果

我们比较了半结构化的 EBMT 系统和基于短语的统计翻译系统 Pharaoh[29]。Pharaoh

系统使用了默认设置：语言模型、排序模型、短语翻译模型和词惩罚模型。Pharaoh 采用与 EBMT 系统相同的双语语料训练翻译模型，然后使用最小错误率训练工具在开发集上来调模型参数的权重[30]。

表 1 给出了翻译评价结果。从翻译结果来看，EBMT 系统的翻译结果在 BLEU 得分上比 Pharaoh 系统高出 17.2%，具有统计显著性。

系统	BLEU 得分
Pharaoh	0.2193±0.0031
EBMT 系统	0.2571±0.0026

表 1. 翻译结果

6 结论

本文介绍了一种半结构化的基于实例的翻译模型。该模型使用 TSC 来表示静态的翻译实例或动态的双语翻译片段。对于待翻译的句子，翻译模型首先得到待翻句子的分析树，接着采用贪心的 TSC-Tree 匹配算法在翻译实例库中找到最佳匹配的 TSC 森林，在此基础上，系统采用线性对数生成模型来得到最后的译文。实验表明，半结构化的基于实例的翻译系统明显优于基于短语的统计机器翻译系统，在 BLEU 得分上取得了 17.2%的提高。

参 考 文 献

[1] Nagao M.. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, In: Proceedings of the International NATO Symposium on Artificial and Human Intelligence, 1984, 173~180

[2] Somers H.. Review Article: Example-based Machine Translation. Machine Translation, 1999, 14(2):113~157

[3] Watanabe H.. A Similarity-Driven Transfer System. In: Proceedings of the 14th International Conference on Computational Linguistics, 1992, 770~776

[4] Poutsma A.. Data-Oriented Translation. In: Proceedings of the 18th International Conference on Computational Linguistics, 2000, 635~541

[5] Al-Adhaileh M.H., Tang E.K., Zaharin Y.. A Synchronization Structure of SSTC and its Applications in Machine Translation. In: Proceedings of the Coling-2002 Post-Conference Workshop on Machine Translation in Asia, 2002, 1~8

[6] Way A.. Machine Translation Using LFG-DOP. R. Bod, R. Scha & K. Sima'an (ed) Data-Oriented Parsing, CSLI Publications, 2003, 359~384

[7] Aramaki E., Kurohashi S.. Example-Based Machine Translation Using Structural Translation Examples. In: Proceedings of International Workshop on Spoken Language Translation, 2004, 91~94

[8] Shieber S.M.. Restricting the Weak Generative Capacity of Synchronous Tree Adjoining Grammar. Computational Intelligence, 1994, 10(4): 371~385

[9] Matsumoto Y., Ishimoto H., Utsuro T.. Structural Matching of Parallel Texts. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, 1993, 23~30

[10] Watanabe H.. A Model of a Bi-Directional Transfer Mechanism Using Rule Combinations. Machine Translation, 1995, 10(4): 269~291

[11] Al-Adhaileh M.H., Tang E.K.. Example-Based Machine Translation Based on the Synchronous SSTC Annotation Schema. In: Proceedings of Machine Translation Summit VII, 1999, 244~249

[12] Aramaki E., Kurohashi S., Kashioka H., Tanaka, H.. Word Selection for EBMT Based on Monolingual

Similarity and Translation Confidence. In: Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, 2003, 57~64

[13] Aramaki E., Kurohashi S.. Example-based Machine Translation Using Structural Translation Examples. In: Proceedings of International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation, 2004, 91~94

[14] Knight K., Hatzivassiloglou V.. Two-level, Many-paths Generation. In: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 1995, 252~260

[15] Kaki S., Yamada S., Sumita, E.. Scoring Multiple Translations Using Character N-gram. In: Proceedings of the 5th Natural Language Processing Pacific Rim Symposium, 1999, 298~302

[16] Callison-Burch C., Flournoy R.S.. A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines. In: Proceedings of MT Summit VIII, 2001, 63~66

[17] Akiba Y., Watanabe T., Sumita E.. Using Language and Translation Models to Select the Best among Outputs from Multiple MT Systems. In: Proceedings of the 19th International Conference on Computational Linguistics, 2002, 8~14

[18] Imamura K., Okuma H., Watanabe T., Sumita E.. Example-based Machine Translation Based on Syntactic Transfer with Statistical Models. In: Proceedings of the 20th International Conference on Computational Linguistics, 2004, 99~105

[19] Carl M., Schmidt P., Schutz J.. Reversible Template-based Shake & Bake Generation. In: Proceedings of MT Summit X Workshop on Example-based Machine Translation, 2005, 17~25

[20] Fellbaum C.. Wordnet: an Electronic Lexical Database. MIT Press, Cambridge, MA, 1998

[21] Lin D.. An Information-theoretic Definition of Similarity. In: Proceedings of the 15th International Conference on Machine Learning, 1998, 296~304

[22] Berger A.L., Della Pietra S.A., Della Pietra V.J.. A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, 1996, 22(1): 39~72

[23] Brown P.F., Lai J.C., Mercer, R.L.. Aligning Sentences in Parallel Corpora. In: Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, 1991, 169~176

[24] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.. Numerical Recipes in C++. Cambridge University Press, Cambridge, UK, 2002

[25] Papineni K., Roukos S., Ward T., Zhu W.. BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, 311~318

[26] 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, 2004, 2051~2054

[27] Stolcke A.. SRILM - An Extensible Language Modeling Toolkit. In: Proceedings of the International Conference on Spoken Language Processing, 2002, 901~904

[28] Collins M.. Head-driven Statistical Models for Natural Language Parsing. [Ph.D. thesis], University of Pennsylvania, 1999

[29] Koehn P.. Pharaoh: A Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In: Proceedings of the 6th Conference of the Association for Machine Translation in the Americas, 2004, 115~124

[30] Och F.J.. Minimum Error Rate Training in Statistical Machine Translation. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 2003, 160~167