doi:10.3772/j.issn.2095-915x.2017.03.007

第一届亚洲翻译研讨会 KyotoEBMT 系统报告

1. 京都大学信息学院 日本 606-8501；2. 科学技术振兴机构 日本 332-0012；
3. 中国科学技术信息研究所 北京 100038；4. 河北地质大学 石家庄 050031

John Richardson[1] Fabien Cromières[2] Toshiaki Nakazawa[2] Sadao Kurohashi[1] 何彦青[3] 刘建辉[4]

摘 要 本文介绍了基于实例的机器翻译引擎 KyotoEBMT 的框架。为了保留语句的非局部结构，我们的系统运用“树到树”的方法对源语言和目标语言都进行了依存句法分析。我们的系统凭借在线的实例匹配和灵活的解码装置确保其最优的翻译效果。实验表明：该系统与当前流行的基于短语的统计机器翻译系统的 BLEU 得分相当。该系统已开源可得。

关键词： 实例 , 树到树 , 句法依存分析 , 机器翻译

中图分类号： G35

KyotoEBMT System Description for the 1st Workshop on Asian Translation

1. Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan;
2. Japan Science and Technology Agency, Kawaguchi-shi, Saitama 332-0012, Japan;
3. Institute of Scientific and Technical Information of China, Beijing 100038, China;
4. Hebei GEO University, Shijiazhuang 050031, China

John Richardson[1] Fabien Cromières[2] Toshiaki Nakazawa[2] Sadao Kurohashi[1]
HE YanQing[3] LIU JianHui[4]

Abstract This paper introduced the KyotoEBMT Example-Based Machine Translation framework. This system used a tree-to-tree approach, employing syntactic dependency analysis for both source and target

作者简介： John Richardson，男，博士，日本京都大学基于EBMT机器翻译系统版权第一持有人，研究方向：机器翻译，Email：john@nlp.ist.i.kyoto-u.ac.jp；Fabien Cromières（1979-），男，研究员，研究方向：机器翻译，Email：{fabien,nakazawa}@pa.jst.jp；Toshiaki Nakazawa（1982-），博士，国立研究开发法人，科学技术振兴机构(JST)研究员，京都大学黑桥&河原研究室成员，研究方向：机器翻译；Sadao Kurohashi（1966-），男，博士，教授，研究方向：机器翻译自然语言处理，Email：kuro@i.kyoto-u.ac.jp。

doi:10.3772/j.issn.2095-915x.2017.03.007

languages in an attempt to preserve non-local structure. The effectiveness of our system was maximized with online example matching and a flexible decoder. Evaluation demonstrated the BLEU scores are competitive with state-of-the-art SMT baselines. The system implementation is available as open-source.

Keywords：Example, tree-to-tree, syntactic dependency analysis, machine translation

1 引言

基于语料库的方法已经成为机器翻译研究的重点所在。如今，一个在源语言和目标语言两端兼用依存句法结构的基于实例的机器翻译平台完美地呈现在我们面前。相对于基于句法的统计机器翻译或基于实例的机器翻译着重研究短语句法树而非依存句法树而言，这一研究范型更倾向于使用“树到串”的方法而非“树到树”的方法。而且，我们对每一种语言都分别使用了依存句法分析器，而不是像(Quirk,2005)[1] 那样从一种语言的依存关系映射到另一种语言。

依存结构信息的“端到端”的使用，可以用于提高翻译实例词对齐的质量，用于制约翻译的规则的抽取并用于指导翻译解码。我们之所以强调更着意于局部语境信息的依存句法结构的重要性，是因为它能对跨越远源的语言对的复杂语句生成流利而准确的翻译。

尽管我们的实验集中在日 - 汉与日 - 英翻译的技术领域，然而我们的系统适用于存在翻译实例和依存句法分析器的所有领域及语言对。

与大多数相似的系统不同，本系统独具之特性在于其不需要预先计算翻译规则。相反，它将每一个输入语句匹配给整个翻译实例库，之后在线提取翻译规则。其优点在于，当创建与合并翻译规则时能获取最大量的信息，同时确保能为输入语句生成类似于既有翻译实例的完美翻译。

本系统主要使用 c++ 开发，并且集成了一个基于 web 的翻译界面以便于使用。web 界面 (见图 1) 也提供了用以进行错误分析的信息，如使用过的翻译实例列表。借助一个基于 Curses 的图形界面，实验时便于进行调优和评估。解码器支持多个线程，且本系统已实现开源可得，源码可以从 http://nlp.ist.i.kyotou.ac.jp/kyotoebmt/ 下载。

2 系统概览

图 2 显示了 KyotoEBMT 翻译流程的基本架构。翻译进程肇始于训练语料之平行语句的句法分析与词语对齐。词对齐使用了基于依存树的贝叶斯（Bayesian）子树对齐模型，其中包含了一个基于树的重排序模型用来获取非局部重排序，这种重排序是那些基于词序列的模型通常不能有效处理的。进而，使用词对齐来构建一个含有“实例”或者“小树（treelets）”的

实例数据库（“翻译记忆库”），这些“实例”或“小树”在解码过程中通过组合形成翻译假设。

执行翻译时，首先对输入语句进行句法分析，然后在实例数据库的条目中搜索能与之匹配的小树。解码器通过调优的对数线性模型进行打分来组合检索到的小树。最后利用重排序算法使用附加的非局部特征（参见第 6 节）在解码器提供的 n-best 列表中再次解码选择最佳翻译。

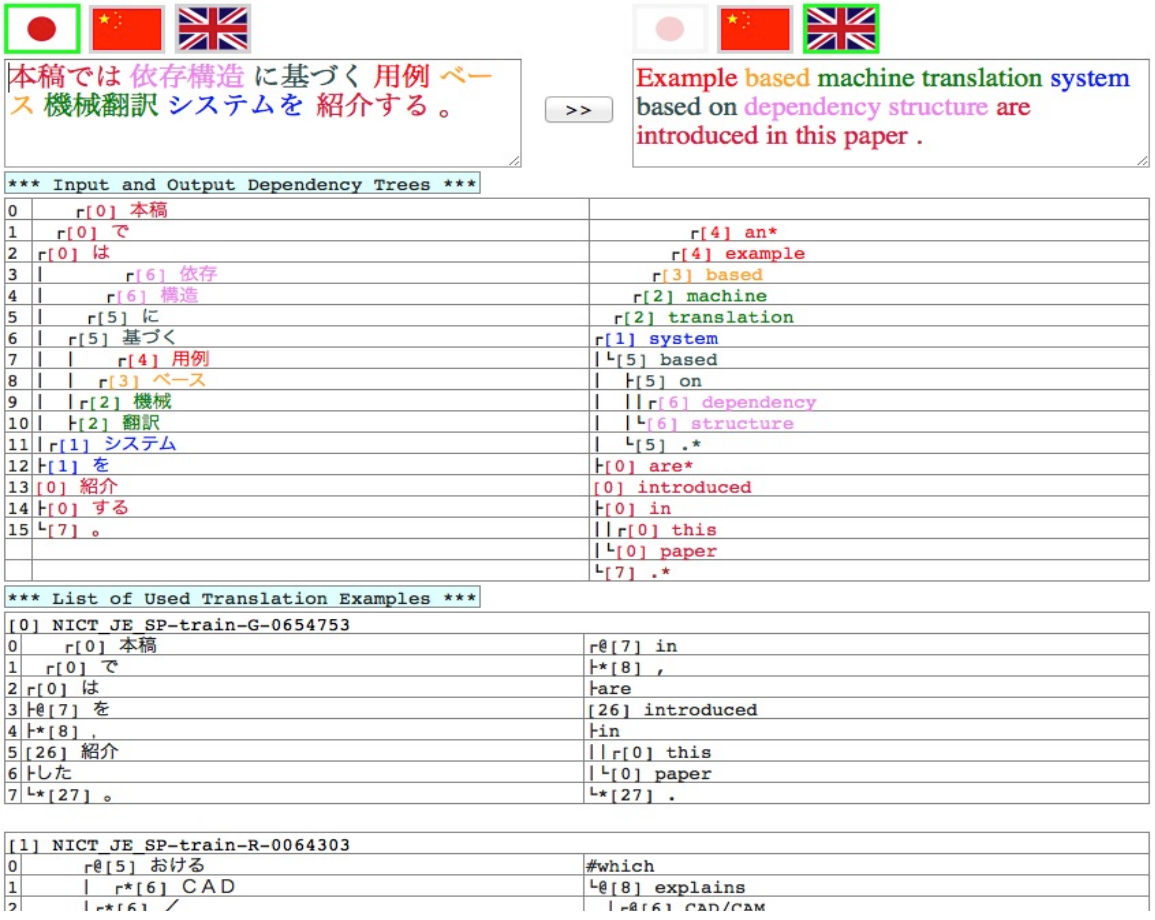

图1　web界面的屏幕截图显示的是日–英的翻译。该接口提供源语言端和目标语言端依存句法树，以及蕴涵词对齐的实例列表。Web 界面有助于简单和直观得进行错误分析，并且可以用作计算机辅助翻译的工具。

为了使得系统对于句法分析错误的更加具有鲁棒性，输入语句被句法分析为所有可能的 k-best 列表，每个句法分析结果都被分别翻译，而所有的翻译在融合之后再做最终的重排序（参见第 7 节）。

第 3 节和第 4 节将会更为详尽地分别解释实例检索和解码步骤。第 5 节则介绍特征的遴选和对数线性模型的调优过程。

图 3 给出了与输入的句法树相匹配的翻译实例进行组合以创建输出语句的过程。

doi:10.3772/j.issn.2095-915x.2017.03.007

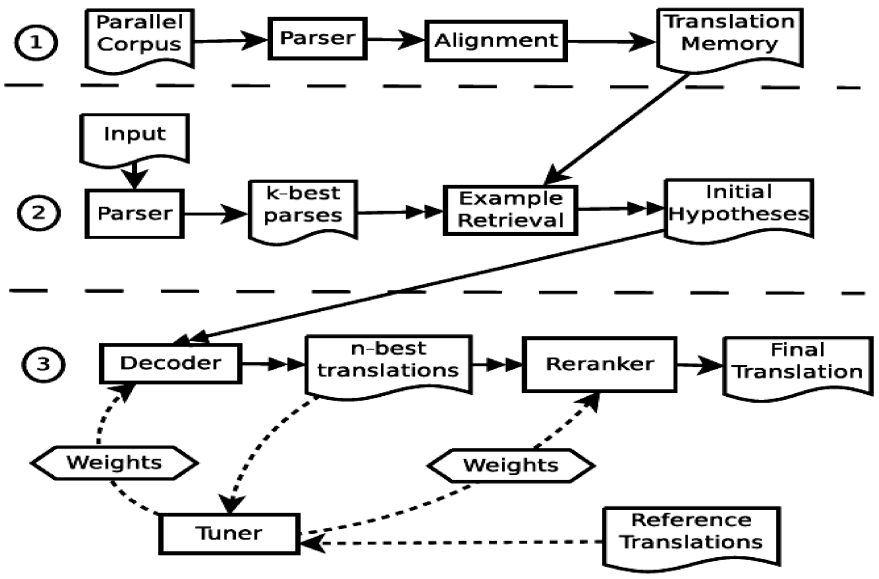

图2 翻译流程大致可以分为3个步骤。步骤1是实例数据库的创建以及平行语料库的训练。步骤2是对一个输入语句进行句法分析从而获取k-best列表，并生成k个初始翻译假设的集合。 步骤3包括解码和重排序。步骤3的修改版本可以用于解码和重排序的权重调优。双箭头表示并行处理输入语句的k个句法分析的每一个结果。

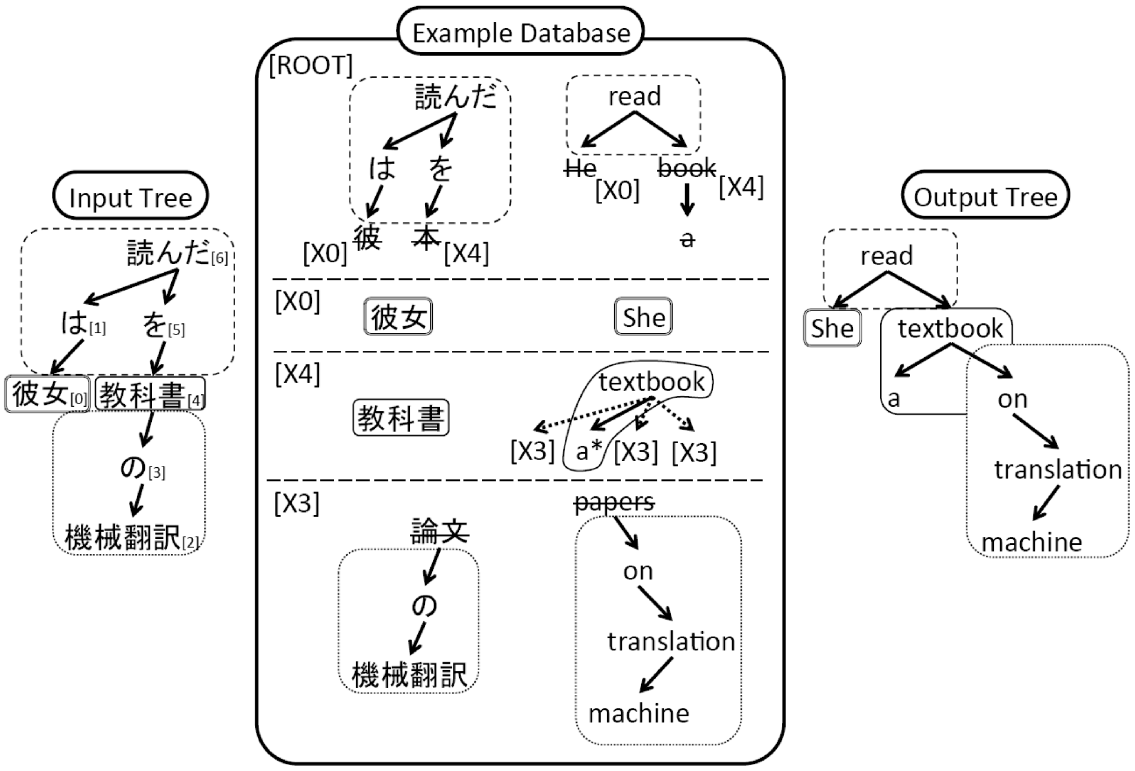

图3 翻译过程。对源语句进行句法分析并检索实例数据库得到匹配的子树。我们从实例中抽取翻译假设，其中包含着可选目标词和每个非终结符的若干位置信息。 例如，包含“textbook“的翻译假设，其非终结符X3（如“a”前的左子节点，“a”后的左子节点或右子节点）具有三个可能的位置。然后在解码期间组合翻译假设。并在解码期间进行词和最终非终结符的位置的选择。

3 实例抽取与翻译假设构建

我们的系统具有一个重要的特性，即不需要提前抽取并存储翻译规则，翻译实例的词对齐是离线完成的。然而，对于给定的输入语句 *i*，无论是寻找能与之部分匹配的实例 *i* 的步骤，还是抽取翻译假设的步骤，都是在线过程。这种方法更加忠实于 Nagao（1984 年）[2] 首倡的 EBMT 方法。人们已经提出了基于短语的系统 [3]，层次化系统 [4] 和基于句法的系统 [5]，但是将 EBMT 整合到基于句法的机器翻译中并不常见。

这种方法有几个优势。首先，提前抽取规则的系统通常会限制抽取规则的规模和数量以免其难于管理，而我们并不需要对翻译假设的规模加以限制。特别是假如输入语句与我们的一个翻译实例相同或极其相似时，我们能够检索到一个完美的翻译。第二个优势在于，我们可以充分利用实例的上下文信息为每个翻译假设分配特征和分数。

我们这一方法的主要缺点在于，从实例数据库中在线检索数量足够多的匹配实例，其计算量比预先计算好规则再进行匹配的成本更高。我们使用文献 [5] 中的技术尽可能高效地执行此步骤。

一旦我们找到一个实例翻译（s; t），其中 s 部分匹配 i，我们即可从中提取一个翻译假设。一个翻译假设被定义为一个通用的翻译规则，将输入语句的 p 部分表示为一个目标语言的小树，其中的非终结符代表句子的其他部分的翻译可以插入的位置。从一个翻译实例中创建翻译假设的方法如下：

1. 我们利用 s 和 t 之间的词对齐关系将 s 中匹配的部分投射到目标端 t。如果 s 和 t 的每个单词都对齐，问题就显得很简单，但通常情况并非如此。因此，我们的翻译假设通常会将一些目标词 / 节点标记为“可选项”：这意味着我们需要在组合时决定是否要将它们添加到最终翻译中。

2. 我们插入非终结符作为投影的子树的子节点。如果 i、s 和 t 具有相同的结构并且完全对齐，问题就很简单，但是事情依旧没有这么容易。结果会导致我们有时需要将每个非终结符插入到若干个可能的位置，在组合的时候再次选择插入的位置。

4 解码

在为输入的句法树找到尽可能多的匹配部分提取到翻译假设之后，我们需要决定如何选择和组合它们。在此我们的方法类似于已经提出的基于语料库的机器翻译方法。我们首先选择一系列特征，并创建一个线性模型，对每个可能的假设组合进行评分（见第 5 节）。然后再尝试找到使该模型得分最大化的组合。

翻译规则的组合受到输入的依存句法树之结构的约束。如果我们只考虑局部特征①，那么一个简单的自底向上的动态规划方法可以有效地找到具有线性 O(|H|) 的复杂度②的最佳组合。

①组合的得分为每个翻译假设的局部得分之和。
②H= 翻译假设集合

doi:10.3772/j.issn.2095-915x.2017.03.007

然而，非局部特征（例如语言模型）迫使我们修剪搜索空间。这种修剪是通过立方剪枝可以有效完成[6]的。我们使用 KenLM③[7]计算目标语言模型得分。通过使用 KenLM 的一些更高级的特征（例如状态规约（state-reduction）[8,9]和休息成本估计（rest-cost estimations）[10]，解码更为有效。与原始的立方剪枝算法相比，我们的解码器设计可以处理任意数量的非终结符。

此外，如第 3 节中所示，最初从实例中抽取的翻译假设有歧义，在使用哪个目标词以及每个非终结符将插入到哪个最终位置无法明确。为了处理这种模糊性，我们使用基于网格的内部表达，就能有效地对它们进行编码（见图 4）。这种网格表示还允许解码器在词的各种形态变化（例如，be/is/are）之间进行选择。我们使用文献 [11] 提供的解码算法。

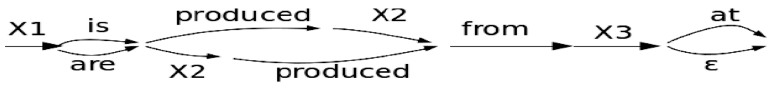

图4 一个被编码为网格的翻译假设。这种表达能让我们有效地处理翻译规则的歧义。值得注意的是，该网格中的每个路径对应于X2的插入位置的不同选择，“be”的语态和“at”的可选插入方式。

5 特征和调优

在解码时，我们使用线性模型来对每种翻译假设的可能组合进行评分。该线性模型是局部特征（每个翻译假设的局部特征）和非局部特征（例如，最终翻译的 5 元语言模型得分）的线性组合。解码器总共考虑 42 个特征的组合，其选择性出示如下：

- 实例惩罚和实例大小
- 翻译概率
- 语言模型得分
- 添加 / 删除可选词语

我们使用包含在 Moses 中的 k-best batch MIRA[12]来估量每个特征的最优权重。

6 重排序

我们使用两个附加的语言模型我们系统的 n-best 输出重新排序：一个是标准的具有 Kneser-Ney 修正平滑的 7 元语言模型，另外一个是递归神经网络语言模型（RNNLM）[13]。RNNLM 模型的训练采用的隐藏层大小为 200，从训练集抽取了 5000 个句子用作验证数据。重排序的时候首先计算 n-best 列表中的每个翻译的各种语言模型得分。第一轮调优使用了一些特征，补充上这些特征后再运行最后一次迭代调优。这次调优的算法和设置与标准调优相同。增加这一步调优的目的在于找到两个附加的语言模型与其他相关特征的最佳组合，诸如句子长度特征，以及解码器内部使用的 5 元语言模型的特征得分。

7 K-best 句法分析

我们发现源语言端的依存句法分析的质量对翻译品质有重要的影响，但是句法分析的错

③ http://kheafield.com/code/kenlm/

误却很难避免。中文的句法分析尤其具有挑战性，我们的中文句法分析器目前仍然存在大量的分析错误。为了缓解这一问题，我们使用输入语句的句法分析 k-best 列表，对每个句法分析结果都并行实施翻译，并在重排序之前进行合并以获取翻译列表。将来，我们打算从现在的多个句法分析 k-best 列表的表达方式转换为更紧凑有效的森林的表达方式。进而，我们将会考虑利用所有翻译实例的多个句法分析结果（而不仅仅是输入语句的句法分析结果）。

8 实验

以下列出了我们使用的依存句法分析器以及它们的参数。括号中的分数为近似的句法分析准确率 (micro-average)，采用了从测试数据中随机抽取子集进行人工评估的方式。在实验中针对不同的领域分别训练了句法分析器。依存句法分析中都使用了 k-best，k 设置为 20。

- 英语：NLParser④ (92%)[14]
- 日语：KNP (96%) [15]
- 汉语：SKP (88%) [16]

8.1 结果

系统进行调优之后，在测试集上进行了翻译效果的评估，结果见表 1。调优是在开发集上进行的，经过 20 多次迭代，n-best 列表中 n=500。

表1 得分

Language Pair	Reranking	BLEU	RIBES	HUMAN
JA-EN	No	20.60	70.12	21.50
	Yes	21.07	69.90	25.00
EN-JA	No	29.76	75.21	33.75
	Yes	310.9	75.96	38.00
JA-ZN	No	27.21	79.13	-0.75
	Yes	27.67	78.83	-0.85
ZH-JA	No	33.57	80.10	6.00
	Yes	34.75	80.26	7.50

9 结论

我们介绍了基于实例的翻译系统 KyotoEBMT，它同时在源语言端和目标语言端使用依存句法分析，并进行了在线实例检索，能够在翻译的时候有效地获取到完整的翻译实例。我们相信使用依存句法分析对于词序差距比较大的语言对获取准确翻译极为重要，特别是在面对大量的长句翻译需求的时候。我们围绕这一设想设计了一个完整的翻译框架，从词对齐、实例抽取再到实例组合，每一步都利用了依存分析树。

参考文献

[1] Quirk C, Menezes A, Cherry C. Dependency

④ http://kheafield.com/code/kenlm/

doi:10.3772/j.issn.2095-915x.2017.03.007

Treelet Translation: Syntactically-informed Phrasal SMT[C]// In Proceedings of ACL 2005, Ann Arbor. 2005: 271-279.

[2] Nagao M. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle[C]// MIT Press, 1984: 351-354.

[3] Callison-Burch C, Bannard C, Schroeder J. Scaling Phrase-based Statistical Machine Translation to Larger Corpora and Longer Phrases[C]// Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005: 255-262.

[4] Lopez A. Hierarchical Phrase-Based Translation with Suffix Arrays[C]// EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic. DBLP, 2007: 976-985.

[5] Cromieres F, Kurohashi S. Efficient Retrieval of Tree Translation Examples for Syntax-based Machine Translation[C]// Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 508-518.

[6] Chiang D. Hierarchical Phrase-Based Translation[J]. Computational Linguistics, 2007, 33(2): 201-228.

[7] Heafield K. KenLM: Faster and Smaller Language Model Queries[C]// The Workshop on Statistical Machine Translation. Association for Computational Linguistics, 2011: 187-197.

[8] Li Z, Khudanpur S. A Scalable Decoder for Parsing-based Machine Translation with Equivalent Language Model state Maintenance[C]// The Workshop on Syntax and Structure in Statistical Translation. Association for Computational Linguistics, 2008: 10-18.

[9] Heafield K, Hoang H, Koehn K, Kiso T, Marcello Federico M. Left Language Model State for Syntactic Machine Translation[C]// In Proceedings of the International Workshop on Spoken Language Translation. 2011: 183-190.

[10] Heafield K, Koehn P, Lavie A. Language Model Rest Costs and Space-efficient Storage[C]// Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012: 1169-1178.

[11] Cromieres F, Kurohashi S. Translation Rules with Right-Hand Side Lattices[C]// Conference on Empirical Methods in Natural Language Processing. 2014: 577-588.

[12] Cherry C, Foster G. Batch Tuning Strategies for Statistical Machine Translation[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2012: 427-436.

[13] Mikolov T, Karafiát M, Burget L, et al. Recurrent Neural Network based Language Model[C]// INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September. DBLP, 2010: 1045-1048.

[14] Charniak E, Johnson M. Coarse-to-fine n-best Parsing and MaxEnt Discriminative Reranking[C]// ACL 2005, Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA. DBLP, 2005: 173-180.

[15] Kawahara D, Kurohashi S. A Fully-lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis[C]// Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics, 2006: 176-183.

[16] Shen M, Kawahara D, Kurohashi S. A Reranking Approach for Dependency Parsing with Variable-sized Subtree Features[J]. 2012.