基于虚拟上下文的统计机器翻译短语表的过滤*

殷乐[1]，张玉洁[1]，徐金安[1]

（1.北京交通大学 计算机学院 北京市 邮编 100044）

摘要：在基于短语的统计机器翻译系统中，自动抽取的短语表中不可避免的包含大量的冗余和错误的短语对，这浪费了解码资源又影响翻译质量。为了缓解这个问题，本文提出一种基于虚拟上下文的过滤短语表的方法。该方法引入虚拟上下文计算短语对的得分增量；并通过计算最大和最小的短语对的得分增量，设计了一种对短语对重排序的过滤策略。我们在 NTCIR-9 的中英数据上进行了验证实验，结果显示，当短语表的规模下降到原来的 47%时，翻译质量的 BLEU 值提高了 0.0005；当短语表的规模下降到原来的 30%时，BLEU 值仅下降 0.0006。实验结果表明，在大规模短语表的过滤中，本文的方法是有效可行的。

关键词：基于短语的统计机器翻译；短语表过滤；虚拟上下文

中图分类号：TP391　　**文献标识码：**A

Phrase table filtration based on virtual context in phrased-based statistical machine translation

Yin Yue[1]，Zhang Yujie[1]，Xu Jinan[1]

（1. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044）

Abstract: In statistical machine translation system, automatically extracted phrase table inevitably contains a large number of errors and redundant phrase pairs, which causes excessive waste of time and space in decoding and affects translation quality. In order to solve this problem, we propose a method for filtering phrase table in which virtual context is introduced to calculate an incremental quantity in score of phrase pair from language model. By considering the maximum and minimum incremental quantity in score from the virtual context, we design a filtering strategy by re-ranking phrase pairs. We conducted experiments on NTCIR-9 Chinese-English data to verify the method. The experimental results show that when the size of phrase table was reduced to 47% of the original, the translation quality was improved slightly; when the size was reduced to 30% of the original, only slight decline occurred in translation quality. The experimental results indicate that this method can effectively filter out the redundant phrase pairs of the phrase table.

Key words: phrase-based statistical machine translation, filter phrase table, virtual context

1 引言

目前层次短语模型是统计机器翻译（SMT）中性能最好的模型之一[1-5]，它可以有效提高翻译质量。但是由于该模型允许在短语中存在变量，造成短语表的规模急剧增大，解码的时间和空间消耗剧增。为了缓解这一问题，本文提出一种基于虚拟上下文的 SMT 短语表过滤方法，可以有效过滤短语表中冗余的短语对，减少解码在时间和空间上的过度消耗。

为了描述我们在短语表过滤上的工作，先简单介绍基于短语的 SMT 中相关的内容。短语表是 SMT 中基本的翻译知识，包含大量的短语对，每个短语对由源语言短语和目标语言短语组成。在构建基于短语的统计机器翻译系统时，需要从平行语料中自动抽取出短语表[6]，并利用目标语言的单语语料库训练目标语言模型。这个构建模型的过程通常称为训练。在翻

* **收稿日期：**　　**定稿日期：**

基金项目：北京交通大学人才基金（KKRC11001532）

作者简介：殷乐（1987——），男，硕士，机器翻译；张玉洁（1961——），女，教授，机器翻译和自然语言处理；徐金安（1970——），男，副教授，机器翻译和自然语言处理。

译时，一个源语言句子首先被分割为短语序列，然后通过短语表，每个短语被翻译成目标语言的短语，最后这些目标语言的短语被重新组合生成一个目标语言句子。这个翻译的过程通常被称作解码，解码的模块被称作解码器。解码器会从候选译文中选取拥有最大概率的译文作为最后的输出。

对于一个源语言句子 $f_1^J = f_1 \cdots f_j \cdots f_J$，一个目标语言句子 $e_1^I = e_1 \cdots e_i \cdots e_I$，F. J. Och and H. Ney[7]提出基于最大熵的 SMT 模型如下。

$$\Pr(e_1^I \mid f_1^J) = p_{\lambda_1}^M(e_1^I \mid f_1^J) = \frac{\exp\left[\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right]}{\sum_{e_1'^I} \exp\left[\sum_{m=1}^{M} \lambda_m h_m(e_1'^I, f_1^J)\right]} \quad (1)$$

其中，$h_m(e_1^I, f_1^J)$ 是特征函数（m=1,2,3…M），λ_m 是特征函数的权重。因为公式中的分母在解码过程中是常数，最优的译文可以通过下面的公式选出。

$$e_1^I = \arg\max_{e_1^I} \{\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\} \quad (2)$$

译文的得分可以通过公式（3）计算得到。

$$Score = \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \quad (3)$$

解码的任务是寻找拥有最大概率的译文，解码已经被证明是一个 NP 完全问题[8]。一个大规模的短语表将会造成解码在时间和空间上的过度消耗，并且影响机器翻译系统的实际应用。特别是将这样一个 SMT 系统移植到如 PDA 一类的移动终端时，几乎是不可能的。

为了解决这个问题，许多研究人员提出了多种过滤短语表的方法，主要集中于两类短语对，即错误的短语对和冗余的短语对。为了过滤短语表中错误的短语对，研究人员提出一种使用对数似然比的过滤方法[9]和使用依存结构限制目标短语数量的方法[10]。为了过滤短语表中冗余的短语对，研究人员则提出一种过滤单调组合短语对的方法[11]和使用对数线性模型过滤组合短语对的方法[12]。

本文中我们提出一种基于虚拟上下文的过滤方法，目标是过滤掉解码过程中几乎不会被使用的短语对。本文中将这种短语对看作是冗余短语对。我们主要集中在两种冗余短语对：同源短语对(ISP）和复合短语对(CPP)。同源短语对指的是那些源语言相同而目标语言不同的短语对。这种短语对意味着同一个源语言短语有多个对应的目标短语，多的情况下会有几十甚至上百个，这种情况下一般会含有冗余短语对。复合短语对是指短语对的源语言短语可以由几个子短语组成，并且在短语表中存在以这些子短语作为源语言短语的短语对。这意味着一个复合短语对可以被几个子短语对替换，出现这种情况时，这些复合短语对中可能含有冗余的短语对。

针对上述的冗余短语对，我们设计实现了短语表过滤器，其处理流程大致如下：首先，过滤器使用对数线性模型计算短语对的得分；然后，过滤器使用虚拟上下文对短语对进行重排序；最后删除掉排名低的短语对。

文章其它部分组织如下，第 2 节详细描述基于虚拟上下文的过滤方法；第 3 节介绍我们的实验和评价结果；最后，在第 4 节给出我们的结论。

2 短语表过滤算法

这一节介绍使用对数线性模型对短语对进行排序的过程，然后详细描述基于虚拟上下文的重排序算法和过滤冗余短语对的策略。

2．1 排序

为了找出几乎不会被解码器使用到的短语对，我们使用和解码器同样的算法评价短语对。我们使用对数线性模型计算短语对的得分，使用的特征包括：翻译概率、词汇化概率、反向翻译概率、反向词汇化概率和语言模型。选用这五个特征的理由是短语对的质量和这些特征密切相关。这些特征的权重通过开发集的数据训练得到。

过滤器对短语对排序的过程如下。

1）选择拥有相同源语言的短语对作为一个集合，用S_i表示；

2）按照公式（3）计算 S_i 中每个短语对的得分，得分最高的短语对表示为 S_i^H，其它短语对表示为 S_i^O，它们的得分分别为 Score (S_i^H) 和 Score (S_i^O)。

2．2 基于虚拟上下文的重排序

通常来说，解码器会选择 S_i^H，而不是 S_i^O。可是 S_i^O 在某些情况下会被解码器选择。出现这种情况的原因可以解释如下。在上文的利用公式（3）的计算中，计算语言模型的得分的过程和解码器的计算过程并不完全相同，因为解码器会使用已生成的译文作为上下文信息计算语言模型。可是，在短语表过滤阶段，没有实际生成的译文供过滤器作为上下文计算。由此，解码器会因为已生成的译文给 S_i^O 一个比 S_i^H 高的得分，导致过滤器和解码器的排序结果不同。为了弥补这一点，我们在过滤器中引入虚拟上下文来计算语言模型并对短语对进行重排序。这种策略可以保证，在重排序后低位短语对在实际解码中基本不会被使用，过滤掉这些短语不会影响翻译质量。

进一步描述的话，解码器中已经生成的译文作为上下文信息在语言模型特征上会产生一个增量，表示为 Δ_{Context}。如果解码器选择 S_i^O 而不选择 S_i^H，是因为 S_i^O 的增量大于 S_i^H，即当公式（4）成立时，解码器会选择 S_i^O。

$$Score(S_i^O) + \Delta_{\text{Context}}(S_i^O) > Score(S_i^H) + \Delta_{\text{Context}}(S_i^H) \qquad (4)$$

在公式（4）中，$\Delta_{\text{Context}}(S_i^O)$和$\Delta_{\text{Context}}(S_i^H)$是 S_i^O 和 S_i^H 分别通过上下文在语言模型特征上获得的增量。

基于以上的考虑，我们设计了一种极端分配上下文的重排序策略，即分配给 S_i^O 最佳上下文使其获得最大增量，而分配给 S_i^H 最差上下文使其获得最小增量。在这种策略中，如果重排序后，S_i^O 的排名依然低于 S_i^H，那么可以说 S_i^O 就很难被解码器选用。这种策略可以简

单表示为S_i^O与最佳上下文对决（vs）S_i^H与最差上下文。

为此，我们引入虚拟上下文模拟S_i^H的上下文，使得$\Delta_{\text{Context}}(S_i^H)$在语言模型上获得最低的得分，同理使用虚拟上下文模拟S_i^O的上下文，使得$\Delta_{\text{Context}}(S_i^O)$在语言模型上获得最高的得分。分别标记它们为$\min \Delta_{\text{Context}}(S_i^H)$和$\max \Delta_{\text{Context}}(S_i^O)$。然后，依据新得分重新排序短语对。如果$Score(S_i^H) + \min \Delta_{\text{Context}}(S_i^H) > Score(S_i^O) + \max \Delta_{\text{Context}}(S_i^O)$，这意味着解码时，在任何上下文的情况下，$S_i^O$都很难被解码器使用到。

短语表的过滤算法如下。

1) 对于一个目标语言短语$W_1\ W_2\ \ldots\ldots W_k$, W_1 和 W_k是短语的边界。如果$W_{x1}\ W_{x2}$ 在2元语言模型中存在并且$\delta(W_{x2}, W_1)=1$, 则把W_{x1}作为目标语言短语的虚拟上下文；如果$W_{x1}\ W_{x2}$在语言模型中存在并且$\delta(W_k, W_{x2})=1$, 则把W_{x2}作为目标语言短语的虚拟上下文。$\delta(x, y)$ 是克罗内克函数，当 x=y时，$\delta(x, y)=1$，否则$\delta(x, y)=0$。除此之外，我们同样考虑了短语中包含变量的情况。给定一个目标语言短语$W_{1\ldots\ldots}W_{m-1}\ X\ W_m\ \ldots\ldots W_k$, X 是一个变量。$W_{m-1}$ 和 W_m 也是短语的边界。如果$W_{x1}\ W_{x2}$ 在语言模型中存在并且$\delta(W_{m-1}, W_{x1})=1$, 则把$W_{x2}$也作为目标短语的虚拟上下文; 如果$W_{x1}\ W_{x2}$ 在语言模型中存在并且$\delta(W_{x2}, W_m)=1$, 则把W_{x1}也作为目标短语的虚拟上下文；
2) 计算$\min \Delta_{\text{Context}}(S_i^H)$:在语言模型中分配一个上下文使得$S_i^H$获得最小得分增量；
3) 计算S_i中的其他短语对的$\max \Delta_{\text{Context}}(S_i^O)$：在语言模型中分配一个上下文使得$S_i^O$获得最大得分增量；
4) 依据获得增量的新得分，对S_i中的短语对进行重排序；
5) 过滤掉排名低于S_i^H的短语对。

在这种极端上下文对比的情况下，排名低于S_i^H的短语对在其他上下文情况下也不会获得更大增量，因此只能排在S_i^H的后面。这意味着在其他情况下，解码器也不会跳过S_i^H而选择排名低于S_i^H的短语对。由此，过滤掉这些短语，译文的质量不会受影响而下降。

下面描述对复合短语对(CPP)的过滤。复合短语对意味着它的源语言短语可以被分解成多个子短语，同时这些子短语的短语对在短语表中存在，我们称这些短语对为子短语对。和同源短语的过滤算法一样，这里也引入了虚拟上下文。过滤算法如下。

1) 计算复合短语对的得分：根据公式（3）先计算一个基础得分，并在语言模型中分配一个上下文使得复合短语对的得分增量最大，二者相加作为复合短语对的得分。
2) 计算子短语对的得分：根据公式（3）先计算一个基础得分，并在语言模型中分配一个上下文使得子短语对的得分增量最小，二者相加作为子短语对的得分。
3) 过滤：如果子短语对的得分之和大于复合短语对的得分，过滤掉复合短语对。

与前面过滤掉的同源短语对的道理相同，在这种极端上下文对比的情况下，复合短语对的得分低于子短语对的得分。这意味着在其他上下文的情况下，解码器只会选择子短语对，而不会选择被过滤掉的复合短语对。因此，在过滤掉复合短语对后，翻译质量不会受影响而下降。

在[11][12]的方法中，过滤复合短语对（$s_1s_2 \rightarrow t_1t_2$）的一个前提条件是短语表中存在短语

对 $s_1 \rightarrow t_1$ 和 $s_2 \rightarrow t_1$，即短语对（$s_1s_2 \rightarrow t_1t_2$）是单调组合的短语对[11]。我们这里只要求在短语表中同时存在源语言短语是 s_1 和 s_2 的短语对.

3 实验

3．1 实验设置

为了验证本文的方法，我们使用一个基于层次短语解码器，在 NTCIR-9 数据上进行了中英方向的实验。NTCIR-9 中英数据的训练集中有一百万句对，测试集和开发集分别有两千句对。

我们在训练集上运行 GIZA++[13]得到双向的单词对齐信息，并使用启发式的方法“grow-diag-final”[1] 改善单词对齐结果；利用单词对齐信息，自动抽取短语表[6]。然后借助工具 SRI language model [14]获得语言模型；通过在开发集上使用最小错误率训练法[15]得到特征的权重。在评测译文的质量时，我们使用 BLEU [16]。

3．1 实验结果

考虑实验的便捷性，我们首先选出源语言短语在测试句子中出现的短语对，作为准备过滤的短语表。

表1. 过滤前后短语表大小和翻译质量的变化

Filtering way	BLEU	PTS	NUM	Reminder
None(baseline)	0.2932	426M	4733693	
ISP	0.2938	230M	2461777	52.42%
CPP	0.2937	309M	3457486	73.03%
ISP&CPP	0.2937	207M	2225644	47.01%

然后我们使用第二节介绍的方法，过滤 ISP 和 CPP。表 1 是过滤前后短语表大小和翻译质量的变化。其中，第一列是过滤的方法（Filtering way），包括 ISP、 CPP 和 ISP&CPP，None (baseline)是过滤前的情况。第二列是翻译质量（BLEU）。第三列是短语表消耗的内存大小（PTS）。第四列是短语表中短语对的数量（NUM）。第五列是过滤后剩余短语对的数量占原短语表的百分比（Reminder）。在过滤 ISP 后，剩余的短语对数量是原来数量的 52.42%，同时 BLEU 值上升 0.0006。在过滤掉 CPP 后，剩余的短语对数量是原来数量的 73.03%，同时 BLEU 值上升 0.0005。在过滤掉 ISP 和 CPP 后，剩余的短语对数量仅占原来数量的 47.01%，同时 BLEU 同样上升 0.0005。实验结果显示同时过滤 ISP 和 CPP 时，效果最好。

为了进一步压缩短语表中的大小，我们考虑在 ISP 过滤中只保留重排序后排名较高的几个短语对。

表2. ISP过滤中保留前5位（TOP 1 –TOP 5）的情况下，短语表大小和翻译质量的变化

Filtering way	BLEU	PTS	NUM	Reminder
None(baseline)	0.2932	426M	4733693	
TOP 5	0.2926	135M	1454004	30.71%
TOP 4	0.2908	124M	1304037	27.54%
TOP 3	0.2896	103M	1112783	23.50%
TOP 2	0.2847	80M	862957	18.23%
TOP 1	0.2716	48M	521787	11.02%

表 2 的结果显示，保留 ISP 排名前五位的短语对获得了最好的实验结果。在过滤掉短语表中大约 70%的短语对后，BLEU 值仅下降 0.0006。该实验结果也显示，保留越少的 ISP 短语对，BLEU 值下降的越快。在我们的实验中，保留排名前五的 ISP 短语对获得了最好的效果，既极大压缩了短语表的规模又没有给翻译质量带来太大的影响。实验证明这种方法在实

际应用中是有意义的。

该方法引入了虚拟上下文计算短语在语言模型特征上产生的得分增量，我们需要注意的是语言模型的稀疏情况。实际解码时，一些 n-gram 在训练得到的语言模型中没有出现，解码器会利用回退或简单平滑的方法处理这些 n-gram。我们的方法在这种情况下的缺陷是实际上可能被解码器使用的短语对在我们的过滤方法中被删除了，这发生在下面两种情形：一是 S_i^H 的实际在语言模型上的增量使其得分低于 $Score(S_i^H)+\min\Delta_{\text{Context}}(S_i^H)$；二是再排序中低于 S_i^H 的短语对中，有的实际在语言模型上的得分增量使其得分高于 $Score(S_i^H)+\min\Delta_{\text{Context}}(S_i^H)$。

关于语言模型的稀疏情况，我们以本实验的数据为对象，用测试集的参考译文的 2-gram 对语言模型进行了测量，结果如表 3 所示。在 2-gram 的情况下，稀疏的情况的百分比只有 11.53%。在大部分情况下，我们的方法计算出的结果应该是正确的。

表3. 测试集的译文2-gram在语言模型中的稀疏情况

2-gram 数量	语言模型中未出现数量	稀疏比
56172	6479	11.53%

4 结论

在这篇文章中，我们提出一种基于虚拟上下文的过滤短语表的方法，通过引入虚拟上下文计算短语对在语言模型特征上获得的最大和最小增量，并设计了对短语对进行重排序的过滤策略。实验结果显示，这种方法可以过滤掉短语表 53%的短语对，同时没有造成翻译质量的下降。在保留重排序后前五名的短语对时，这种方法可以过滤掉 70%的短语对，同时 BLEU 值仅有 0.0006 的极微小的下降。实验证明这种方法可以有效过滤掉短语表中冗余的短语对，极大压缩短语表的规模。

在以后的工作中，我们将尝试融合其它信息进一步提升这种方法的有效性。

参考文献

[1] Philipp Koehn, Och, F.J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (pp. 127—133)

[2] R. Zens, F.J. Och, H. Ney, 2002. "Phrase-Based Statistical Machine Translation". In: M. Jarke, J.Koehler, G. Lakemeyer (Eds.): KI - 2002: Advances in artificial intelligence. 25. Annual German Conference on AI, KI 2002, Vol. LNAI 2479, pp. 18-32, Springer Verlag, September 2002

[3] Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas, pp. 115-124.

[4] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In Proceedings of ACL 2005, pages 263–270, 2005.

[5] D. Chiang. Hierarchical phrase-based translation. Computational Linguistics, 33(2):201–228, 2007.

[6] F. J. Och and H. Ney. The alignment template approach to statistical machine translation [J]. Computational Linguistics, 30(4):417-449. June 2004

[7] F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, July

[8] K. Knight. 1999. Decoding complexity in word replacement translation models. Computational Linguistics, 25(4).

[9] Wu, Hua and Haifeng Wang. 2007. Comparative Study of Word Alignment Heurist Based SMT. In Proc. of Machine Translation Summit XI, pages 507-514.

[10] L. Shen, J. Xu, and R. Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In Proceedings of ACL-08: HLT, pages 577–585, Columbus, Ohio, June 2008.

[11] Z. He, Y. Meng, Y. Lj, H. Yu, and Q. Liu. Reducing SMT rule table with monolingual key phrase. In Proceedings of the ACL-IJCNLP 2009 Conference, pages 121–124, Singapore, August 2009.

[12] Seung-Wook Lee, Dongdong Zhang, Mu Li, Ming Zhou, and Hae-Chang Rim. 2012. Translation model size reduction for hierarchical phrase-based statistical machinetranslation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics(Volume 2: Short Papers), pages 291–295, Jeju Island,

[13] Korea, July. Association for Computational Linguistics. F. J. Och and H. Ney. Improved statistical alignment models. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pages 440–447, 2000.

[14] A. Stolcke. Srilm – an extensible language modeling toolkit. In Proceedings of the International Conference on Spoken language Processing, volume 2, pages 901–904, 2002

[15] F. J. Och. Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pages 160–167, 2003.

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, 2002.