移动终端机器翻译设备的解码定点化方法

李响[1]，徐金安[1]，刘群[2]，吕雅娟[2]，姜文斌[2]
1 北京交通大学计算机与信息技术学院 北京 100044
2 中国科学院计算技术研究所 北京 100190
E-mail:lixiang@ict.ac.cn

摘 要： 面向移动终端的统计机器翻译设备的需求越来越多，但翻译速度受无浮点运算单元的处理器制约。本文提出了一种对统计机器翻译的解码定点化方法，缓解了无浮点运算单元的处理器对翻译速度的影响。基于 PC 和移动终端的实验表明，定点解码器在保证翻译质量的情况下，其定点运算速度较浮点运算提高 135.6%。因此，本方法可以有效地提高浮点运算能力薄弱的移动终端统计机器翻译设备的翻译速度。

关键词： 统计机器翻译，定点化，移动终端

A Fixed Point Decoding Approach for Mobile Machine Translation Terminals

Xiang Li[1]，Jinan Xu[1]，Yajuan Lü[2]，Qun Liu[2]，Wenbin Jiang[2]
1 School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044
2 Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190
E-mail:lixiang@ict.ac.cn

Abstract: The demand for statistical machine translation (SMT) on mobile terminals is increasing, but the translation speed is restricted by the processor without floating point unit (FPU). This paper proposes an approach to switch floating point to fixed point for decoder of SMT system on mobile terminals, which reliefs the effect of the processor without FPU on translation speed. The experiments based on PC and mobile terminal show that this approach assures the quality of translation and the speed of fixed point operation is 135.6% faster than the speed of floating point operation. Therefore, this approach can efficiently increase the translation speed of SMT system on mobile terminals with weak ability in floating point operation.

Key words: statistical machine translation, fixed point, mobile terminals

1 引言

统计机器翻译为克服语言障碍提供了一个解决方向，而嵌入式技术的飞速发展使移动终端可以运行复杂的统计机器翻译系统，可以预见，具有高效统计机器翻译功能的移动终端将成为有此需求人士的必备工具。

目前移动终端上的机器翻译系统主要采用基于规则的方法，这种方法需要针对每种语言设计翻译规则，而规则获取难，不易于扩充；然而，基于统计的机器翻译不依赖语言学知识，模型能够较好的支持多语言互译，但翻译模型庞大，翻译耗时长，消耗系统资源较多[1][2]。

移动终端硬件性能的大幅提升使统计机器翻译在移动终端上实现成为可能，然而，无浮点运算单元的处理器影响了需要大量浮点运算的统计机器翻译的解码速度。对此，本文提出了将解码运算定点化的方法。实验结果表明，在保证较好翻译质量的情况下，本方法可以有效地提高统计机器翻译在移动终端上的翻译速度。

2 统计机器翻译系统

本文以 Xiong et al. (2006)的系统 Bruin[3]为实验系统，重新实现了一个定点化解码器。

Bruin 系统主要由以下三个部分组成：

(1) 随机括号转录语法(BTG)[4]，并用对数线性形式的多种特征对规则赋予权重；

(2) 基于最大熵的调序模型，其特征通过双语训练集自动学习；

(3) 采用 Beam Search 的 CKY 类型的解码器。

下面，我们主要介绍通过 BTG 对翻译过程的建模实现，如下式所示。

$$A \rightarrow [A^1, A^2] \tag{1}$$

$$A \rightarrow \langle A^1, A^2 \rangle \tag{2}$$

$$A \rightarrow x / y \tag{3}$$

规则(3)将源短语 x 翻译为目标短语 y，并生成一个块 A，规则(1)和(2)以正向或反向顺序将两个连续的块合并为一个更大的块。

Bruin 采用对数线性模型计算规则概率，从而构建一个随机 BTG。对于规则(1)和(2)，指定的概率定义如公式(4)。

$$\Pr^m(A) = \Omega^{\lambda_\Omega} \cdot \Delta^{\lambda_{LM}}_{p_{LM}(A^1, A^2)} \tag{4}$$

其中，Ω 表示通过最大熵调序模型得到的 A^1 和 A^2 的重调序得分，λ_Ω 表示 Ω 的权重；$\Delta_{p_{LM}(A^1, A^2)}$ 表示根据调序结果得到的两个块的语言模型分数增量，λ_{LM} 表示 $\Delta_{p_{LM}(A^1, A^2)}$ 的权重。

规则(3)的概率定义如公式(5)。

$$\begin{aligned} \Pr^l(A) = {} & p(x|y)^{\lambda_1} \cdot p(y|x)^{\lambda_2} \cdot p_{lex}(x|y)^{\lambda_3} \cdot \\ & p_{lex}(y|x)^{\lambda_4} \cdot \exp(1)^{\lambda_5} \cdot \exp(|y|)^{\lambda_6} \cdot p_{LM}^{\lambda_{LM}}(y) \end{aligned} \tag{5}$$

其中 $p(\cdot)$ 表示短语双向翻译概率，$p_{lex}(\cdot)$ 表示词汇化双向翻译概率，exp(*1*)表示短语惩罚，exp(|*y*|)表示单词惩罚，λ_s 表示特征权重。

3 数值理论基础

下面主要介绍与定点化过程密切相关的数值理论基础。

3.1 浮点数值基础

在 IEEE-754 浮点标准中[5]，浮点数是将特定长度的连续字节分割为特定长度的符号域 S，指数域 E 和尾数域 M，则浮点数表示形式如图 1。

图 1 浮点数值类型表示形式

另外，IEEE-754 标准明确规定了以下两种基本浮点格式。

(1) 单精度浮点格式

N 共 32 位，其中 S 占 1 位，E 占 8 位，M 占 23 位。

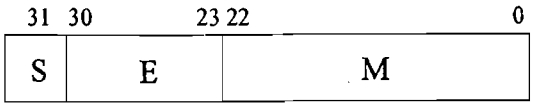

图2 单精度浮点格式

(2) 双精度浮点格式

N共64位，其中S占1位，E占11位，M占52位。

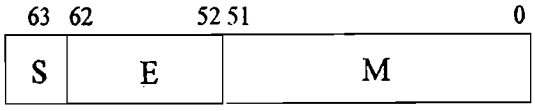

图3 双精度浮点格式

3.2 定点数值基础

定点数据类型约定参与运算的数值的小数点隐含在某一固定位置上，而在对小数点位置做出选择后，运算中的所有数均应统一为定点数据类型，在运算中不再考虑小数问题。

定点数是将特定长度的连续字节分割为特定宽度的符号域S，整数域IWL和小数域FWL，其表示形式如图4。

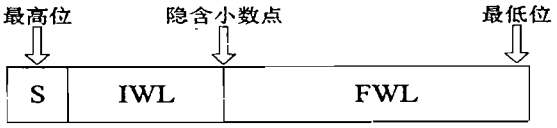

图4 定点数值类型表示形式

4 解码器定点化

无浮点运算单元的处理器对浮点运算的处理的主要有以下三种传统方法：

(1) 定义变量为浮点型，高级程序语言自动调用运行时函数来处理浮点运算，消除了定点处理器和浮点处理器的区别，用户编程工作量少，但编译代码庞大，运算速度慢；

(2) 定义变量为整形，采用放大倍数表示浮点数，该方法虽然简单，但缺乏灵活性；

(3) 采用内核中的非定义指令异常中断处理方式，浮点指令被截获并由浮点模拟器(NWFPE或FastFPE)模块来执行，这种方式将会导致CPU频繁产生异常，降低了运算速度。

为了解决上述传统方法中的不足，本文提出了软件方式的定点化处理方法，提高无浮点运算单元的处理器对浮点运算的处理速度。

4.1 定点化格式设计及算法

数的定标是通过程序员来决定小数点在定点数中的位置，从而确定小数的范围和精度。Q格式是一种数的定标格式，其小数部分位数已经设定，例如，Q15表示该定点数有15位小数。因此，针对移动终端上统计机器翻译解码，我们采用补码形式的Q格式定点数来处理对浮点运算，可以通过改变定标值，使程序更加灵活，适应不同的处理器和统计机器翻译解码器。

将一个基本浮点数转换为 Qn 格式的 W 位定点数，所执行的操作如下：

(1) 获取存储浮点数的连续字节内容于一个 W 位整型变量；

(2) 根据IEEE-754标准对基本浮点数的存储格式定义，获得浮点数的符号，指数和尾数；

(3) 根据指数对尾数进行缩放，然后根据 Qn 格式，将缩放后的尾数转换为相应的定点数，超过定点字长的位采用截断方式处理。

最后，我们以实数1.1234为例说明其转换方式。其单精度浮点数存储格式如图5。

S	E	M
0	0111111	00011111100101110010010

图 5 实数 1.1234 的单精度浮点存储格式

其中：

符号：S＝0，表明该浮点数为正数；

指数：E＝0x7F，实际指数 E' ＝E－Bias＝127－127＝0；

尾数：M＝0x8FCB92，实际尾数 M' ＝M | 0x800000＝0x8FCB92。

下面采用 *Q8* 格式的 short 类型变量替换浮点数，则实数 1.1234 的定点存储格式如图 6。

S	IWL	FWL
0	0000001	00011111

图6 实数1.1234的Q8格式定点存储格式

4.2 解码器定点化方案

用定点运算模拟浮点运算是整个解码器定点化过程中的重点，需要预先知道解码过程中参与浮点运算的浮点变量的动态范围，然后用一定精度的定点类型变量来表示它们，再通过定点运算法则模拟实现原来的浮点运算。

下面，我们介绍对 Bruin 系统解码器的定点化工作。

(1) 由于解码器采用的语言模型直接调用 SRILM[6]接口，我们无法对其定点化，因此，我们只对 SRILM 产生的短语概率结果进行定点化；

(2) 由于解码器使用了语言模型、短语惩罚等九个特征，并为每个特征赋予权重，并采用式(6)计算当前短语翻译评分，因而我们便对其中的特征值和权重值分别进行定点化。

$$score = \arg\max_{e} \sum_{m=1}^{M} \lambda_m h_m(e, f)$$

经过如图 7 的对解码器的定点化处理后，定点运算在功能上完全和被替换的浮点运算一样，只是在精度上略逊于浮点运算。当把这样的替换应用到整个的解码器代码空间后，并且能保证代码运行的结果在误差允许的范围之内时，相当于消除了所有的浮点运算，此时 Bruin 系统解码器也已经定点化。

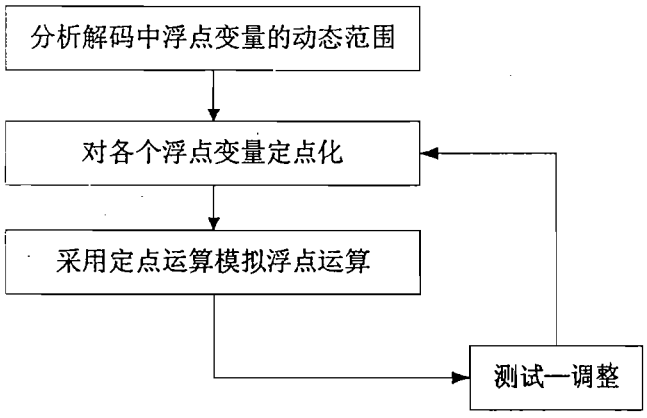

图 7 定点化流程

5 实验

下面，我们将分别在 PC 和移动终端上进行定点解码器和浮点解码器的性能对比实验。

5.1 PC 实验

本实验均是在汉英方向上进行，并且使用 FBIS(共 239K 双语对)作为训练集，NIST 2002 汉英测试数据作为开发集，NIST 2005 汉英测试数据作为测试集。

我们使用 SRILM 工具包对训练语料的目标端训练了一个 4 元语言模型，并采用 Kneser-Ney[7] 平滑方法。同时，我们使用大小写不敏感的 BLEU[8]来衡量翻译质量。

表 2 的 PC 实验结果表明，定点解码器可以保证较好的翻译质量，而速度稍慢于浮点解码器。可以预测，将定点解码器移植到无浮点运算单元的移动终端中，翻译速度将会有较大提升。

表 1 PC 实验平台配置信息

处理器	内存容量	操作系统
Quad-Core AMD Opteron Processor 8347 HE, 1.9 GHz	60 GB	Red Hat Enterprise Linux AS, X64

表 2 PC 实验结果

	翻译时间	BLEU
浮点解码器	8978. 74s	0. 3035
定点解码器	9066. 57s	0. 3035

由于统计机器翻译系统对内存要求较大，限于硬件条件，我们采取在移动终端上对定点和浮点数值性能进行对比试验，从侧面验证我们的统计机器翻译定点化解码器的解码速度。

5.2 移动终端实验

我们的移动终端定点和浮点数值性能对比实验分别对定点数值和浮点数值进行 10,000,000 次的循环运算，其中运算类型采用统计机器翻译解码中运算频率较高的乘法和累加运算。

表 4 的实验结果表明，定点数值运算性能较浮点运算性能提高 135.6%，从侧面可以验证，将本文设计的定点解码器移植到无浮点运算单元的移动终端中，会有效提高统计机器翻译的解码速度。

表 3 移动设备实验平台配置信息

处理器	内存容量	操作系统
Intel ARM920T PXA27X, 312MHz	64MB	Windows Mobile 6.0 Standard

表 4 移动终端实验结果

	运行时间
浮点运算	14.75s
定点运算	6.26s

6 总结

本文提出的面向移动终端的统计机器翻译解码定点化方法，有效地提高了其在浮点运算能力薄弱的移动终端上的翻译速度，同时保持了较好的翻译质量。另外，本方法也适用于语音识别，语音翻译等相关领域，具有较高的实用价值。

下一步，我们希望通过分析统计机器翻译解码器各模块的性能消耗并针对通用移动终端平台，提出一些其他的优化方法，提高移动终端统计机器翻译设备的翻译速度，改善用户体验。

参 考 文 献

[1] 徐金安，机器翻译展望，2010 海峡两岸信息科学与信息技术学术交流会议，2010，秦皇岛

[2] 刘群，统计机器翻译综述,中文信息学报，2003，17(4):1-12

[3] Deyi Xiong, Qun Liu, Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, p. 521-528, 2006.

[4] Dekai Wu. Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora. In *Proceedings of IJCAI 1995*, p. 1328-1334, 1995.

[5] IEEE Standards Board and ANSI. IEEE Standard for Binary Floating-Point Arithmetic, 1985, IEEE Std 754-1985.

[6] Andreas Stolcke. SRILM—An Extensible Language Modeling Toolkit. *Proc. Intl. Conf. on Spoken Language Processing*. v. 2, p. 901-904, 2002.

[7] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology, 1998.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, p. 311-318, 2002.