doi:10.6043/j.issn.0438-0479.201811006

一种可扩展的深度神经网络机器翻译 Service 架构

张 巍[1*],林飞飞[1],梁镇爽[2],黄 振[2]
(1.中国海洋大学信息科学与工程学院,山东 青岛 266100;
2.中译语通信息科技(青岛)有限公司,山东 青岛 266061)

摘要:提出了一种可扩展的基于深度神经网络方法的在线翻译系统架构方法,采用 GPU 和 CPU 混合解码的后端部署方法来提高系统的并发能力,降低系统延迟.实验结果表明,所提出的系统架构方法相比于只使用 GPU 或 CPU 架构,系统并发能力更强,而响应延迟相对较低.同时系统的架构方法可以方便地扩展到多服务器架构中,整体上提高系统的性能.

关键词:神经机器翻译;在线翻译;混合解码
中图分类号:TP 391 **文献标志码**:A **文章编号**:0438-0479(2019)02-0184-05

神经机器翻译(NMT)[1]作为一种新兴的翻译技术,近年来取得了极大的发展.尽管 NMT 方法存在着诸如词汇表受限等缺点,但是与传统的统计机器翻译(SMT)[2-4]方法相比,在翻译的流利度与可读性方面取得了明显的进步,译文质量也有了显著的提高,目前已经成为了机器翻译研究的主流.

互联网的迅速发展为机器翻译的研究提供了巨大的应用空间.在工业界、谷歌、百度等公司将机器翻译作为研究重点的同时,依托其强大的硬件资源对外提供多种语言间的翻译服务.与此同时,诸多垂直行业对定制的机器翻译系统有着巨大的需求.传统的基于 SMT 的翻译系统使用 CPU 进行解码,而 NMT 系统则往往在 GPU 上解码,并且有着低延迟的优势.将 SMT 系统的架构方法直接应用于 NMT 系统是不合适的,而有关 NMT 系统在实际应用中的研究并没有见到公开发表,因此基于 NMT 的在线翻译系统研究有着强烈的实用价值.

本文中提出了一种可扩展、可定制、并发能力强的 NMT 线上系统架构方案.该系统采用了 GPU 和 CPU 混合解码的后端架构方法,充分利用了系统的硬件资源来提高并发数,同时又兼顾 GPU 低延迟和 CPU 高并发的优点.测试结果表明,系统的并发处理能力得到了提高,在整体上实现了较高并发和较低延迟,能够可靠稳定地对大量并发的用户请求进行处理.

1 NMT

NMT 作为一种不同于统计方法的翻译技术,很好地解决了统计翻译方法面临的局部特征严重影响译文质量、远距离依赖信息无法被学习和使用的问题[5-7].端到端的 NMT 采用编码-解码的框架来实现自然语言之间的转换[8],通过基于长短记忆[9]和注意力机制[10-11]的递归神经网络[12]来有效捕捉长距离依赖信息.

由于 NMT 方法相对 SMT 方法表现出来了优良的性能,学者们已经逐渐将研究的重点从传统的 SMT 方法转到了基于深度学习的 NMT 方法[13].同时,诸如 Nematus,OpenNMT,DL4MT 等优秀的 NMT 工具被开发了出来,人们可以根据需要来选择合适的语料训练 NMT 模型.

作为机器翻译的重要应用,在线翻译系统的研究有着重要的应用意义.然而谷歌机器翻译等 NMT 系统无法满足大量企业对定制机器翻译的需求.由于 NMT 解码方法与 SMT 方法有着很大的不同,传统的基于 SMT 的系统架构方法难以直接适用于 NMT 服

收稿日期:2018-11-05 **录用日期**:2019-01-10
* **通信作者**:weizhang@ouc.edu.cn
引文格式:张巍,林飞飞,梁镇爽,等.一种可扩展的深度神经网络机器翻译 Service 架构[J].厦门大学学报(自然科学版),2019,58(2):184-188.
Citation:ZHANG W,LIN F F,LIANG Z S,et al. A scalable deep neural network machine translation service[J]. J Xiamen Univ Nat Sci,2019,58(2):184-188. (in Chinese)

http://jxmu.xmu.edu.cn

务器,因此基于 NMT 的翻译系统的架构方法需要进一步的研究. 下一节,将对提出的 NMT 系统架构方法进行详细的介绍.

2　深度 NMT 系统框架

2.1　NMT 系统整体架构

本文中提出的基于 NMT 的在线翻译系统框架如图 1 所示,系统分为显示层、中间层和解码层 3 个部分.

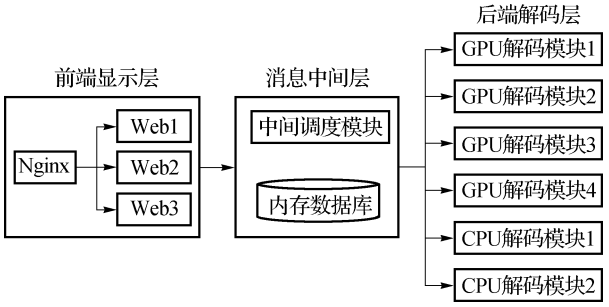

图 1　NMT 系统整体架构图

Fig. 1　Architecture of NMT service

显示层采用了 Nginx 和 Web 服务器组合的方式,来提高系统的扩展性和稳定性. 当大量的用户同时发出请求时,Nginx 一方面将请求等概率地转发给内部的服务器,另一方面通过设置最大访问数的方法来实现对过量的并发请求的处理,从而避免系统故障.

中间层分为中间调度模块和内存数据库模块两部分. 用户请求信息交由中间调度模块处理,而携带的数据则通过内存数据库来实现高速传输. 在提高中间层调度模块的控制管理能力的同时,保证了数据传输的可靠与效率.

解码层由 GPU 解码模块和 CPU 解码模块组成. 通过采用多模型并发和混合解码的方法来提高系统的并发处理能力,同时降低系统的响应延迟,使系统在整体上能够实现高并发与低延迟.

2.2　关键技术

2.2.1　消息中间层

消息中间层主要有两个任务:1) 在大量的前端用户请求与采用混合架构的多个后端解码模块之间建立稳定的一对一连接,使其可以进行可靠的消息传递,2) 对 GPU 和 CPU 混合解码模块进行监控,确定空闲模块的优先级,实现最优的资源分配. 为了提高中间层的效率和可靠性,本文中将中间层分成了中间调度模块和数据传输模块两个部分,下面介绍具体的实现方法.

中间调度模块由 ZeroMQ 来实现. 中间层与前后端的交互情况如图 2 所示,通过使用 ZeroMQ 中经典的请求/应答模式和 REQ-ROUTER 的套接字组合来实现消息的可靠传递. 客户端的 REQ 与 ROUTER 建立稳定的一对一连接,传递消息并等待后端的响应. ROUTER 套接字通过维护一个散列表确定客户端的身份,并将用户请求排队,依次分配后端解码资源,并将响应后的请求信息转发给对应的客户端. 为了保证程序的稳定性,中间层采取了负载均衡的策略来对客户端的请求进行处理,当请求数超过系统处理能力时,接收到的消息将被舍弃.

中间调度模块与后端解码层的监控与调度是通过建立一个空闲状态队列的方式来实现的. 当解码程序空闲时会向调度层发送状态,进入空闲队列,当被中间层调度后,从空闲队列中移除. 由于后端采用混合 GPU 和 CPU 解码的架构方式,调度层会优先调度 GPU 解码资源,以保证系统资源的最佳利用.

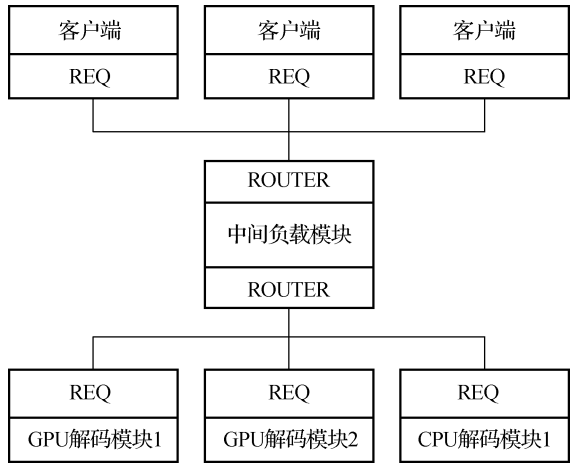

图 2　中间层请求-应答模式

Fig. 2　Request-response mode of middle layer

数据传输模块由内存数据库 Redis 实现,Redis 数据库将数据以 key-value 的方式临时存储在内存中,相比于其他数据库更加便于操作,同时数据存取效率更高,可以有效减少数据的传输延时.

Web 服务后台处理程序与中间层的交互流程如图 3 所示. 首先,根据用户请求发出的时间和 IP 生成唯一的标识,并将标识与数据分别作为 key 与 value 的值存入数据库. 其次,通过 ZeroMQ 将请求交由中间层的调度模块进行处理. 最后,响应后的请求,经后台处理程序从数据库中读取出对应的数据,显示到页面之中.

http://jxmu.xmu.edu.cn

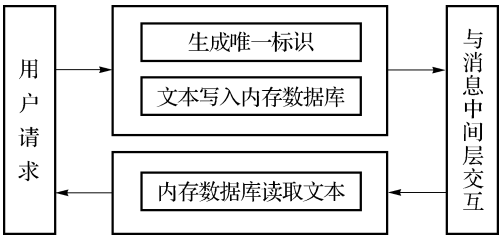

图 3 中间层消息处理流程

Fig. 3 Message handling flow of middle layer

2.2.2 后端解码层架构

NMT 系统可以在 GPU 和 CPU 上进行解码，GPU 的解码效率往往远远高于 CPU 的解码效率，但是却无法像在 CPU 上一样实现较高的并发. 当 GPU 解码模型并发数较高时，系统会变得不稳定. 为了使系统在整体上实现较高并发和较低延迟，本文中仍旧以 GPU 并发为主，处理大多数的用户请求，保证系统的低延迟. 与此同时，采用少量模型在 CPU 上解码来增大并发数，使系统整体实现较高并发与较低延迟.

解码层面临的主要问题是如何确定最佳的 GPU 和 CPU 解码模型的并行数，以实现系统并发处理能力的最大化，使在兼顾 GPU 低延迟与 CPU 高并发优势的同时，又保证系统的整体解码效率. 本文中以系统是否能稳定地对大量并发的用户请求进行处理为标准，来确定最佳的 GPU 和 CPU 解码模型的并行方案，具体设置的测试方案需要根据具体的硬件环境来确定. 在追求最大并发数的同时，优先保证 GPU 的并发数，以确保系统的整体解码效率.

3 实 验

3.1 实验说明

3.1.1 实验环境

实验采用的 NMT 解码系统为 Github 上开源的 Amunmt 神经机器翻译系统，下面用“Amunmt”表示，NMT 模型为爱丁堡大学使用 nematus 训练的英德翻译模型.

为了观测系统在较低配置的机器上的运行情况，以确保服务器在进行多机扩展后能够提供较高性能的服务，选择 Centos7.0 操作系统. CPU 2 核 4 线程，内存 8 GB，显卡 GTX 1060 显存 6 GB.

3.1.2 测试方法

下文中的测试均使用 Tsung 进行的多用户并发测试. 本文中将 Tsung 开始生成用户，直到所有的用户都得到响应的时间定义为一个完整的并发响应时间.

3.1.3 Amunmt 参数测试

实验中，通过多个 Amunmt 模型分别在 GPU 和 CPU 上并行解码的方式来实现混合解码. 首先，需要对单个 Amunmt 模型的配置文件进行设置，以确保任意一个 Amunmt 模型运行时都能达到最大的解码效率，Amunmt 配置文件的主要参数是 CPU 和 GPU 的线程数；然后，以单个 Amunmt 模型解码时的解码速率为参考，分别测试配置文件中 CPU 和 GPU 线程数在不同情况下系统的解码用时情况. 测试数据为平均词数为 25 词的 10 句英文文本.

实验中只考虑 GPU 线程数为 0 和 1 的情况，分别使用 200 用户并发来测试 CPU 线程数在不同参数下的系统响应时间，结果如表 1 所示.

表 1 CPU 线程测试结果

Tab. 1 Result of CPU threads test

CPU 线程数	响应时间/s	
	不使用 GPU	使用一个 GPU
1	3 510	126
2	1 968	344
3	1 872	394
4	1 810	597

根据测试结果，确定实验中单个 Amunmt 模型在进行 CPU 和 GPU 解码时配置文件的参数 CPU 线程数/GPU-线程数分别为 2/0 和 0/1.

3.1.4 混合解码并行数测试

为了确定混合架构下解码层的 GPU 和 CPU 的并行方案，同时与只利用 GPU 和只利用 CPU 并行方案进行对比，本文中测试了系统稳定解码时，后端 GPU 和 CPU 的最大并发情况.

根据测试的硬件条件，确定单用户进行翻译的数据为：平均词数为 50 词，总句数为 100 句的英文文本. 测试文本中句子信息的直方图和正态分布如图 4 所示，句子的单词数整体分布在 45 到 55 之间.

经测试，只利用 GPU 的最大并发数为 4，而只利用 CPU 的最大并发数为 5，混合架构的最大并发数为 5，具体有 GPU∶CPU 为 3∶2 和 4∶1 两种情况. 由此可见，混合架构可以在一定程度上提高系统的并发

http://jxmu.xmu.edu.cn

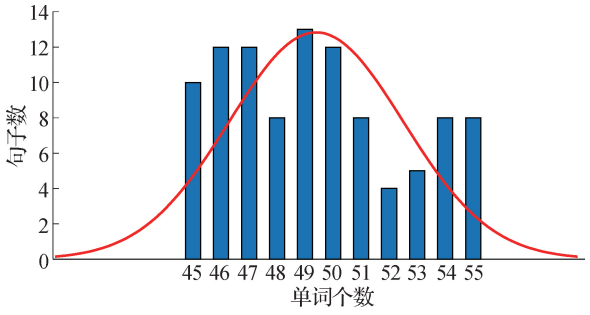

图 4　测试文本句子分布

Fig. 4　Sentence distribution of text

数,同时 GPU 解码模型承担了大部分的解码任务,使系统的整体解码效率并未受到较大影响.

3.2　系统稳定性测试

本实验中,对系统在不同用户并发情况下的系统响应时间变化情况进行了测试与分析. 图 5 显示的是在 100 到 1 000 用户并发下系统响应时间的变化情况. 由图 5 可知随着用户并发数的增加,系统响应时间呈现线性增长,说明混合架构的后端部署方案在 1 000并发范围内可以稳定地进行程序的解码.

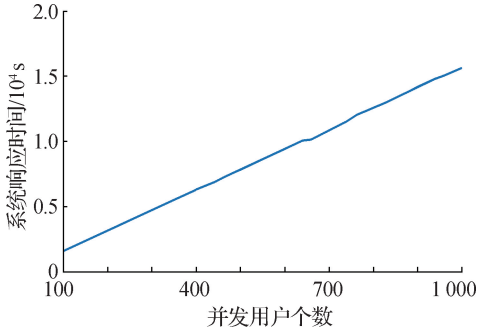

图 5　混合架构系统情况下的响应时间变化情况

Fig. 5　Response time of mixed architecture system

3.3　系统并发能力测试

本节主要观察了混合架构下系统在多用户并发访问的情况下系统平均响应时间的变化情况. 为了观察低用户并发访问时系统的响应情况. 将用户并发数少于 10 时系统的平均响应延时单独进行了记录,如表 2 所示. 为了保证系统整体解码效率,测试中采用稳定的混合架构(GPU∶CPU=4∶1),即解码层并发数为 5 的方案. 系统在 200 用户并发内系统的平均响应时间变化情况如图 6 所示.

由于在进行压力测试时也将 Tsung 程序的启动和加载等时间记入在内,所以导致少量用户并发时,系统的平均响应延时较高. 由表 2 可以看出,随着并发数的变化,系统的平均响应时间逐渐减小,并且呈现起伏状态,说明在用户并发数较低时,CPU 解码模型对系统整体响应延时的影响较大.

表 2　1～10 用户并发测试表

Tab. 2　1～10 user concurrent test results

用户并发数	平均响应延时/s
1	7.60
2	4.05
3	2.75
4	2.18
5	4.04
6	3.13
7	3.07
8	2.54
9	2.46
10	1.85

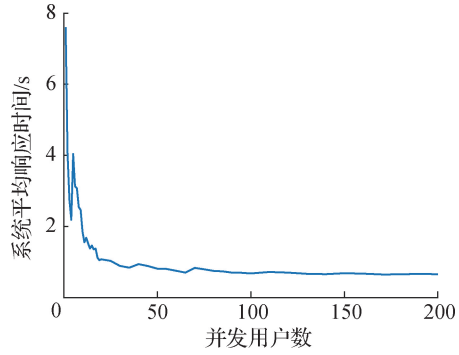

图 6　混合解码并发测试图

Fig. 6　Concurrency tests of mixed decoding system

由图 6 的测试结果可以看出,用户并发数低于 20 时,系统平均响应时间相对较高;当用户并发数超过 20 之后,系统的平均响应时间趋于稳定,并逐渐接近只利用 GPU 解码时的系统响应延时 0.64 s.

从以上的测试结果可以看出,尽管在用户并发访问数较低时,混合解码的后端架构方法受 CPU 模型解码速率较慢的响应较大,但是整体上来说本系统实现了较高并发和较低延迟. 相比于只利用 CPU 或只利用 GPU 的后端架构方案,混合的方法运行更加稳定,并发能力更强,平均延迟较低.

4　结　论

本文中提出了一种基于 NMT 的翻译系统架构方法,它采用 GPU 和 CPU 混合解码的方式来提高

http://jxmu.xmu.edu.cn

系统的并发能力.该方法与只利用 GPU 架构方法相比,整体效果上表现得更好,并发能力更强,延迟较低.

从实验结果可以看出,在单台服务器上系统可以通过混合解码的方式充分利用系统资源,提高并发能力.同样的特点可以扩展到多服务器的架构之中,有效降低服务器的部署成本,实现高并发与低延迟.

参考文献:

[1] SUTSKEVER I, VINYALS O, LE Q. Sequence to sequence learning with neural networks[C] // Proc of the 28th NIPS. New York: Curran Associates Inc, 2014: 3104-3112.

[2] BROWN P, DELLA PIETRA S, DELLA PIETRA V, et al. The mathematics of statistical machine translation: Parameter estimation [J]. Computational Linguistics, 1993, 19(2): 263-311.

[3] CHIANG D. A hierarchical phrase-based model for statistical machine translation[C] // Proc of the 43rd ACL. Stroudsburg: ACL, 2005: 263-270.

[4] OCH F J, MARCU D. Statistical phrase-based translation [C]// HLT-NAACL. New York: ACM, 2003: 48-54.

[5] 刘洋. 基于深度学习的机器翻译研究进展[J]. 中国人工智能学会通讯, 2015(10): 28-32.

[6] 王海峰, 吴华, 刘占一. 互联网机器翻译[J]. 中文信息学报, 2011, 25(6): 72-80.

[7] 庞斌. 机器翻译——从统计学方法到神经网络[J]. 研究与探讨, 2016(12): 296-297.

[8] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[EB/OL]. [2018-10-01]. https: // arxiv. org/abs/1409. 0473.

[9] GRAVES A. Long short-term memory[M] // Supervised sequence labelling with recurrent neural networks. Berlin Heidelberg: Springer, 2012: 1735-1780.

[10] JUNCZYS-DOWMUNT M, DWOJAK T, HOANG H. Is neural machine translation ready for deployment? A case study on 30 translation directions[EB/OL]. [2018-10-01]. https: // arxiv. org/abs/1610. 01108v2.

[11] LUONG T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: [s. n.], 2015: 1412-1421.

[12] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[13] 刘洋. 神经机器翻译前沿进展[J]. 计算机研究与发展, 2017, 54(6): 1144-1149.

A scalable deep neural network machine translation service

ZHANG Wei[1*], LIN Feifei[1], LIANG Zhenshuang[2], HUANG Zhen[2]

(College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China;
Global Tone Communication Technology (Qingdao) Co., Ltd., Qingdao 266061, China)

Abstract: Neural network machine translation, which is a new machine translation method, has become the mainstream of machine translation research. In this paper, we propose an extensible online translation system architecture based on deep neural network, which builds the system backend through the method of GPU and CPU mixed decoding to improve the concurrency ability of the system, and reduce system delay. Experimental results show that the proposed system architecture method is effective. Compared to pure GPU or CPU architecture, the system has higher concurrency ability and the response delay is relatively low. At the same time, the architecture can be extended to the multi-server architecture and improve the performance of the system further.

Keywords: neural machine translation service; online translation; hybrid decoding

http: // jxmu. xmu. edu. cn