北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis

基于 MapReduce 的中文词性标注 CRF 模型并行化训练研究

刘滔　雷霖　陈荦†　熊伟

国防科学技术大学电子科学与工程学院，长沙　410073; † E-mail: luochen@nudt.edu.cn

摘要　针对条件随机场模型面对大规模数据传统训练算法单机处理性能不高的问题，提出一种基于 MapReduce 框架的条件随机场模型训练并行化方法，设计了条件随机场模型特征提取及参数估计的并行算法，实现了迭代缩放算法的并行。实验表明，所提出的并行化方法在保证了训练结果正确性的同时，大大减少了训练时间，性能得到了较大提升。

关键词　词性标注；条件随机场; MapReduce；并行

中图分类号　TP391

A Parallel Training Research of Chinese Part-of-Speech Tagging CRF Model Based on MapReduce

LIU Tao, LEI Lin, CHEN Luo†, XIONG Wei

School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073;
† E-mail: luochen@nudt.edu.cn

Abstract　Conditional random field (CRF) model bears a major drawback of low training efficiency for large-scale data processing. A parallel method of conditional random field model training based on MapReduce is proposed to solve the problem mentioned above. The method designs parallel algorithm for feature selection and parameters estimation of conditional random field model to achieve a parallel iterative scaling algorithm. Eexperiment result shows that the method improves the performance and reduces time cost significantly.

Key words　part-of-speech (POS) tagging; conditional random field(CRF); MapReduce; parallel

词性标注是自然语言处理的基础性课题，是进行更高级处理的基础技术，在机器翻译、语音识别、文本校对、信息检索等自然语言处理的很多领域都发挥着重要的作用，并在各个自然语言处理系统中得到广泛应用。词性标注就是为句中每一个词指派一个合适的词性，其难点在于识别兼类词，即对于有多个词性的词怎么给出一个符合上下文环境的词性。

中文词性标注主要有基于规则的方法和基于统计的方法[1]，基于规则的方法费时费力且规则有限，带有很强主观性，同时对于不规范的句子很难识别，词性标注准确率不高。因此，目前主要采用基于统计的方法，包括隐马尔科夫模型[2]、最大熵模型[3]和条件随机场模型[4]。隐马尔科夫模型要求严格的独立假设，只能利用到上文的信息而不能利用下文的信息；最大熵模型又存在转移偏置；相比之下，条件随机场能够充分利用上下文信息以及各种特征，提高识别能力。但是，传统条件随机场方法的缺点在于，模型训练时间很长，对于规模大的训练语料甚至无法训练。

目前，MapReduce 在自然语言处理中的研究应用比较热门，Lin 等 [5]探讨了基于 MapReduce 的数

新闻出版重大科技工程项目(1041STC40889/01-2)和 863 计划(2011AA120300)资助
收稿日期: 2012–05–30；修回日期: 2012–08–14；网络出版时间：2012-10-26 17:55
网络出版地址：http://www.cnki.net/kcms/detail/11.2442.N.20121026.1755.017.html

据密集型文本的处理方法，如 EM(Expectation maximization)算法和隐马尔科夫模型的 MapReduce 并行化；张佳宝[6]提出了基于 MapReduce 的条件随机场模型并行化命名实体识别技术。

本文针对条件随机场模型，面向大规模的训练语料，采用基于 MapReduce 的开源 Hadoop 并行处理框架[8]，提出一种对模型训练过程进行并行化处理的方法。实验表明，该方法在保证词性标注准确性的同时，大大降低了训练过程所需的时间，性能得到了较大提升。

1 背景知识与问题描述

1.1 MapReduce 框架

MapReduce[7]是由 Google 提出的用于处理和生成大数据集的编程模型，它隐藏了任务调度、机器容错、数据划分、负载均衡、机器通信等并行化细节，使得编程人员能够简单方便地使用 MapReduce 来解决问题。

在 MapReduce 计算过程中，将计算分成 Map 和 Reduce 两个阶段，两阶段的输入输出都是 key/value 对：Map 阶段，处理输入 key/value 对，并产生中间 key/value 对；Reduce 阶段，根据中间 key/value 对的 key 合并对应的 values，产生最后的输出 key/value 对。Map 和 Reduce 的过程如下。

$$\text{Map}: <k_1, v_1> \rightarrow \text{list}<k_2, v_2>,$$

$$\text{Reduce}: <k_2, list(v_2)> \rightarrow <k_3, v_3>。$$

MapReduce 的执行框架如图 1 所示。

1.2 条件随机场

条件随机场[4](conditional random field, CRF)是一种无向图模型，是在给定待标记的观察序列的条件下计算整个标记序列的联合概率分布，是 Lafferty 等在 2001 年提出，其模型思想主要来源于最大熵模型。

令 $G=(V,E)$ 表示一个无向图，$Y=(Y_v)_{v\in V}$，Y 中元素与无向图 G 中的顶点一一对应。在条件 X 下，随机变量 Y_v 的条件概率分布服从图的马尔科夫属性：$p(Y_v \mid X, Y_w, w\neq v)=p(Y_v \mid X, Y_w, w\sim v)$，其中 $w\sim v$ 表示 (w,v) 是无向图 G 的边。(X,Y) 称为一个条件随机场。

已知条件概率 $p(y \mid x,\lambda)$ 的形式化公式为

$$p(y \mid x,\lambda)=\frac{1}{Z(x)}\exp(\sum_k \lambda_k F_k(y,x)), \tag{1}$$

其中归一化因子 $Z(x)$ 的表达式为

$$Z(x)=\sum_y \exp(\sum_k \lambda_k F_k(y,x))。 \tag{2}$$

对于 CRFs 概率模型而言，参数估计的任务是从相对独立的训练数据中估计式(1)的参数 λ_k 的值。在条件随机场模型的参数估计常用的模型训练的方法有 GIS(generalized iterative scaling)算法[8]和 IIS(improved iterative scaling)算法[9]。

1.3 中文词性标注条件随机场模型

条件随机场能够充分地利用上下文信息从而达到良好的标注效果，目前实际应用中最常用的是一阶链式结构[10]的 CRF 模型，即线性链结构(linear-chain CRFs)。本文采用 CRF 模型的线性链结构，具体结构如图 2 所示。

条件随机场模型需要考虑 3 个关键的问题：特征函数的选取、参数估计以及模型推断。

1.3.1 特征函数选取

特征函数的选取是 CRF 模型需要考虑的首要

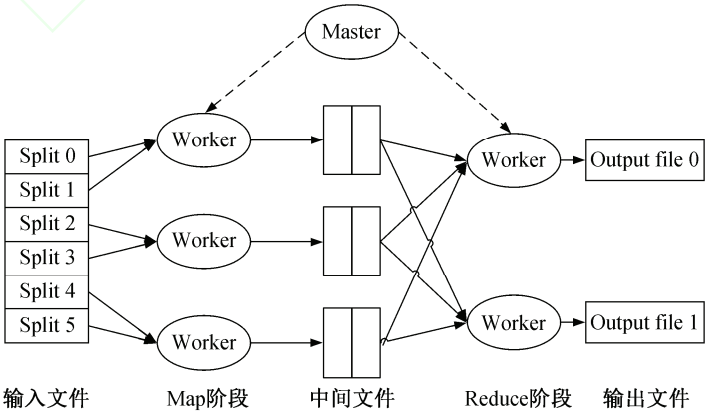

图 1 MapReduce 执行框架

Fig. 1 MapReduce framework

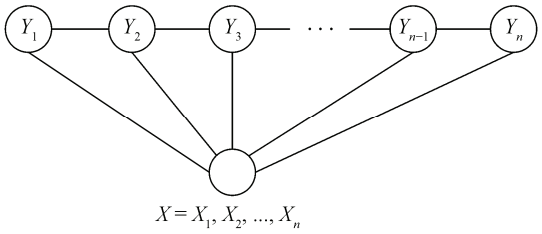

图 2　CRF 模型线性链结构
Fig. 2　Linear-chain CRF

问题。对于词性标注而言, 首先考虑相当一部分词只有一个词性。对于这类词, 可以仅用当前词及其词性建立特征函数即可。以“中国”这个词为例, 只存在一种词性, 其特征函数如下:

$$f_a(x,y)=\begin{cases}1, & x=\text{中国}, y=\text{NN},\\ 0, & \text{其他}。\end{cases}$$

但是词性标注的难点在于兼类词的识别标注: 对于存在多种词性的词, 如果只用当前词及其词性建立特征函数, 我们就只能根据模型计算出的条件概率大小来判断词性, 判断的结果都是统计概率最大的词性, 这不符合实际。为了实现兼类词的识别及标注, 就需要结合上下文的信息。本文考虑当前词前后一个或多个词对当前词的影响建立特征函数。以“工作”这个词为例, 它存在两种词性: 名词和动词。考虑前后一个词, 对于“我的工作很轻松。”这句话, 提取出“工作”做为名词时的部分特征函数如下:

$$f_a(x,y)=\begin{cases}1, & x=\text{工作}, y=\text{NN},\\ 0, & \text{其他},\end{cases}$$

$$f_a(x_{-1},x,y)=\begin{cases}1, & x_{-1}=\text{的}, x=\text{工作}, y=\text{NN},\\ 0, & \text{其他},\end{cases}$$

$$f_a(x,x_1,y)=\begin{cases}1, & x=\text{工作}, x_1=\text{很}, y=\text{NN},\\ 0, & \text{其他}。\end{cases}$$

同样, 作为动词也可以提取出相应的特征函数, 特征模版如表 1。

只考虑当前词及其词性获取特征函数, 特征函数数量不超过词的数量和可能的词性的乘积, 数据量不大; 考虑前后一个词及其词性建立特征函数, 获得的特征函数数量将急剧增长; 进一步考虑更多的上下文信息, 即使用前后更多的词及其词性来建立特征函数, 特征函数数量将成指数增长。

1.3.2　参数估计

参数估计是条件随机场模型最重要最关键的问题。参数估计就是从训练数据中学习式(1)的参数, 即 $\lambda=(\lambda_1,\lambda_2,\cdots\lambda_n)$, 通常通过极大似然估计来实现。

表 1　特征模版
Table 1　Feature templates

特征	说明
W_0	当前词
W_1	当前词的后一个词
$W_0\ W_1$	当前词和后一个词
W_{-1}	当前词的前一个词
$W_{-1}W_0$	当前词和前一个词
$W_{-1}\ W_0\ W_1$	前一个词、当前词和后一个词

假设给定训练集 $D=\{(X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n)\}$, 其中 X_i 表示输入的词序列, Y_i 表示对应的词性序列。对训练集 D 使用极大似然估计可以求解参数值, 但是求解的复杂度极高, 一般不直接求解。通常可以采用迭代缩放(GIS 算法和 IIS 算法)的方法来估计极大似然参数, 但 GIS 算法和 IIS 算法复杂度仍然较高, 需要极长的训练时间, 甚至无法训练。

1.3.3　模型推断

模型推断就是在训练好模型参数 $\lambda=(\lambda_1,\lambda_2,\cdots\lambda_n)$ 后, 使用模型对未标记文本进行词性标注。目前已有相关的研究[6]使用 MapReduce 并行命名实体识别 CRF 模型的推断过程。本文主要研究训练过程的并行化, 不再对模型推断过程进行并行化实现。

2　基于 MapReduce 的 CRF 模型训练并行化

词性标注的条件随机场模型训练分成两步: 1) 特征函数获取; 2) 模型参数估计。

2.1　特征函数获取的并行

特征函数获取过程和词频统计的过程有很大的相似。Map 函数的输入为训练语料, key 为文件每行的行偏移, value 为每行数据。value 包含的两个元素分别是词以及对应的词性。对于只使用当前词及其词性提取特征函数的情况, 将 value 包含的词, 对应词性以及特征函数初始参数 $\lambda_k=1.0$ 以及分隔符作为输出 key, 使用 1 作为输出 value。Combine 函数和 Reduce 函数是相同的简单求和。最后输出结果 key 就是特征函数及其参数 λ_j, 输出结果的 value 是对应特征函数经验分布的数学期望。

中间 key/value 对通过 partition 函数分配到不同

的 Reduce 函数，同时按 key 值进行排序。为了使相同词的 key/value 对集合到一起，需要使用自定义的 partition 函数而不是默认的 partition 函数，按第一个字母分块的 partition 函数即求 key 值的第一个字母的 ASCII 码除以 Reduce 个数的余数。

为了方便下一步的计算，根据式(1)和(2)中计算条件概率需要使用到相同 x 的所有数据，我们按 x 将所有数据分组集合到一起。

为了更好地找到不同数据分组之间的间隔位置，我们加入一个特殊的 key/value 对作为间隔，在获得每一行数据时，将 value 中的词加上特殊标识作为中间结果的 key，同时使用 1 作为中间结果的 value，经过排序后，该 key/value 对将处于每一个数据分组的末尾。

对于双词语(多词语)特征函数的获取，我们需要同时对上下两行(多行)的数据进行处理，处理的过程和单词语特征函数获取过程相似。

2.2 模型参数估计的并行

在 2.1 节，我们已经获取了特征函数，接下来需要对每个特征函数的权值 λ_k 进行计算。迭代缩放是一种通过更新规则以更新模型中的参数，通过迭代改善联合或者条件模型分布的方法，其更新规则如下:

$$\lambda_j \leftarrow \lambda_j + \delta\lambda_j \text{。}$$

更新值 $\delta\lambda_j$ 使得新的参数值 λ_j 比原来的值参数 λ_j 更接近极大似然值，通过一次次的迭代最终使得参数值满足要求。

整个迭代过程不适合并行，因为每一步迭代需要使用到上一步的结果。但是，因为大规模 CRF 模型训练中，特征函数的数量极大，可以将迭代的每一步进行并行化，并行计算不同的特征函数参数，从而减少参数估计的时间。迭代过程其中一步的并行过程如下：

```
建立新 List
定义分组间隔标志符 sEND
while key/value 存在
    read key/value 对
    key/value 对插入 List
    if key 中含有 sEND
     计算处理 List 并输出
     建立新 List
    end if
end while
```

根据 2.1 节，获得数据已分组并给出了特殊间隔标识，因此上述并行过程中，每次读取一个 key/value 对的数据存入 List 中，直到满足第 6 行条件，即找到数据分组末尾的特殊间隔标识，对 list 中的数据进行处理计算并更新 λ_k，输出处理的结果，开始处理下一对数据。

2.3 并行性能分析

通常情况下，并行计算同时处理 p 个任务，最大可能达到的加速比为 p。但是由于 MapReduce 并行过程中通信需要时间，所有子任务时间不可能全部同时完成等原因，在 MapReduce 并行计算中，加速比实际并不能达到 p。

考虑负载均衡尽量使所有子任务基本同时完成，读取数据的就近原则使得通信时间尽量短，备份机制防止因子任务失败而造成更大时间开销，这些都是 MapReduce 框架自身的优化策略。计算节点数将会是影响加速比的最大因素，越多的计算节点才能使得加速比的最大上限越大。对于 MapReduce 而言，确定合适 map 数 m 和 reduce 数 r 使得并行性能达到最佳。假设 map 和 reduce 阶段顺序计算的时间分别为 t_1 和 t_2，map 阶段同时运行的最大 map 数为 m，reduce 阶段 reduce 数为 r，理论上总的时间花销为 $t \approx \frac{t_1}{m} + \frac{t_2}{r}$。

目前 Hadoop 框架根据数据量大小 S 和计算节点数 N 自动确定同时运行的 map 数量 m，因此节点数 N、数据量大小 S 以及 reducer 数量 r 将是影响并行性能的最大因子。下面，通过实验在这 3 方面对并行性能进行具体分析。

3 实验与分析

实验环境为 IBM HPC 计算平台，其中包括 2 个 IBM X3650M3 管理节点，28 个 2 路 8 核 IBM Blade Center HS22 刀片服务器以及 4 个 IBM X3550M3 机架服务器组成 32 个计算节点，所有计算节点和管理节点通过一台 40 GB QDR Infiniband 网络交换机互联，运行 Red Hat Enterprise Linux 5.5 和 Apache Hadoop-0.21[①]。

在实验中，我们需要用到大规模的语料，但是

① http://hadoop.apache.org/

由于条件限制, 没有规模足够大的标准标记语料。我们采用新闻语料, 并经过 ICTCLAS2011[①]分词系统进行分词等预处理作为实验训练文本。语料大小为 4 GB。实验通过 3 个方面来分析 CRF 模型并行训练的性能: 训练样本大小 S、计算节点数 N 和 reducer 数 r, 同时一个实验验证并行化过程的正确性。

3.1 实验 1: 不同训练样本大小的效果

固定 N=16, r=12, CRF 模型训练迭代计算 20 次。实验 1 分析不同的训练样本大小 S 对模型并行化训练的影响, 实验结果如图 3。

图 3 中不同大小的训练样本对训练时间的影响情况如下。获取特征函数的时间随着样本规模增长而成线性增长, 而迭代计算时间随着样本规模增长而增长, 但增长幅度越来越小。主要时间花销是迭代计算。随着训练样本规模的增长, 特征函数需要从所有样本中去获取, 从而时间花销随着样本增长而成线性增长。随着训练样本增加, 更多的特征在前面文本已经存在, 从而增加的特征数量慢慢变少, 而迭代过程是计算特征函数的参数, 因此时间花销增长幅度越来越小, 在样本充分的时候最后将趋于一个平稳。

实验采用 GB 规模的训练样本进行训练, 但是, 随着文本不断增大, 时间花销的增加是可以预计的, 最后直到迭代计算的时间不再增加, 增加的时间都是获取特征函数部分的时间。到了迭代计算时间不再增长时, 增加样本没有新的特征函数, 增加训练样本大小就已经没有意义。

3.2 实验 2: 不同计算节点数的效果

固定 S=4 GB, r=12, CRF 模型训练迭代计算 20 次。实验 2 分析不同的计算节点数对模型并行化训练的影响, 结果如图 4 所示。

随着计算节点数的增加, 训练时间变短。20 个计算节点, 根据配置每个节点可以同时处理 4 个子任务, 整个集群可以同时运行 80 个 map 或 reducer 子任务, 对于 4 GB 的训练样本, 完全能够满足计算要求。超过 20 个节点后, 随着计算节点数的增加对计算性能影响很小。

3.3 实验 3: 不同 reducer 数的效果

固定 N=16, S=4 GB, CRF 模型训练迭代计算 20 次。实验 3 分析不同的 reducer 数对模型并行化训练的影响, 结果如图 5 所示。

随着 reducer 数的增加, 训练时间变短。输出结果即 CRF 模型较大, 多个 reducer 可以提高效率。reducer 数为 32 时, Hadoop 的公平调度技术将确保每一个节点同时运行一个 reducer, 而不是采取“就近计算”的原则, 从而部分通信代价将大于计算代价造成性能下降, 执行时间略有上升。

3.4 实验 4: 验证并行化的正确性

由于数据量剧增的问题, 目前没能对大规模训

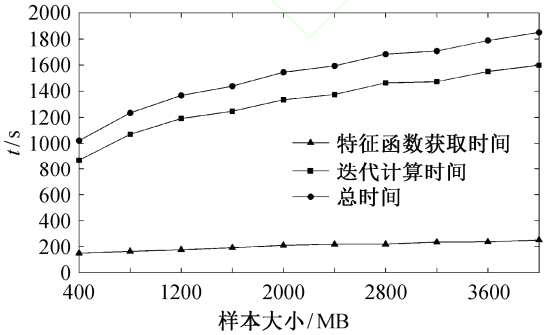

图 3 训练样本的影响

Fig. 3 Influence of the training samples

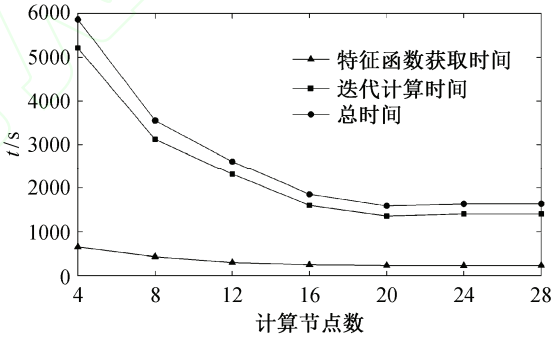

图 4 计算节点数的影响

Fig. 4 Influence of the computing node number

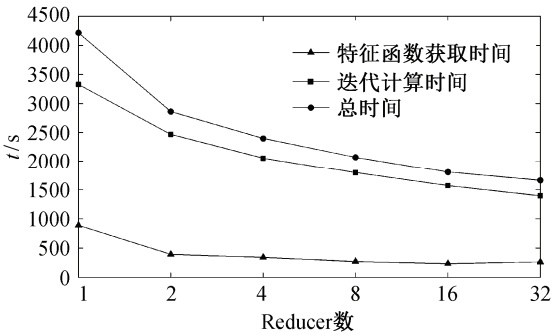

图 5 Reducer 数的影响

Fig. 5 Influence of the reducer number

① http://ictclas.org/

练样本训练出的模型进行验证。在小规模样本情况下, 使用前后一个词的特征函数模版, 采用 3000 K 的样本作为训练样本, 在单机以及 Hadoop 平台上训练所得的模型, 对 60 K 的测试样本进行词性标注, 单机和并行所得的结果一致, 并行过程没有改变算法的正确性。实验结果如表 2 所示。

表 2　词性标注结果

Table 2　Results of part-of-speech tagging

	单机		并行	
	查准率/%	查全率/%	查准率/%	查全率/%
兼类词	96.32	93.21	96.32	93.21
非兼类词	94.34	90.45	94.34	90.45

4　结语

本文基于 MapReduce 计算模型, 详细分析了 CRF 模型并行化训练方法, 并给出了基于 MapReduce 的 CRF 模型并行化训练的实现, 分析 CRF 模型训练并行化的性能。实验表明, 采用 MapReduce 并行化可以很好地解决大规模训练语料的训练问题, 大大缩短了训练时间。在后续工作中, 会考虑将大规模训练样本训练的模型用于模型推断的并行方法, 以及改进目前广泛应用的训练工具 CRF++的现有方法并扩展至多机分布式平台, 进一步提高性能。

参考文献

[1] 洪铭材. 基于条件随机场(CRFs)的中文词性标注方法. 计算机科学, 2006, 33(10): 148–155

[2] 王敏. 基于改进的隐马尔科夫模型的汉语词性标注. 计算机应用, 2006, 26(增刊 2): 197–198, 207

[3] Ratnaparkhi A. Maximum entropy models for natural language ambiguity resolution[D]. Philadelphia: University of Pennsylvania, 1998

[4] 于江德. 基于条件随机场的汉语词性标注. 微电子学与计算机. 2011, 28(10): 63–66

[5] Lin J, Dyer C. Data-intensive text processing with MapReduce. San Francisco: Morgan and Claypool Publishers, 2010

[6] 张佳宝. 基于 Hadoop 的并行化命名实体识别技术研究与实现 // 全国计算机安全学术交流会论文集. 2010, 25: 126–130

[7] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM, 2008, 51(1): 107–113

[8] Darroch J N, Ratcliff D. Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics, 1972, 43(5): 1270–1480

[9] Berger A. The improved iterative scaling algorithm: a gentle introduction: technical report. Pittsburgh: School of Computer Science, Carnegie Mellon University, 1997

[10] 韩雪冬. 条件随机场理论综述[EB/OL]. (2010–01–13)[2011–09–23]. http://www.paper.edu.cn/ index.php/default/releasepaper/content/201001–479