**基于条件随机场的中文短文本分词方法**

**本文创新点：1.比较了多种标记方式**

**2.在特征选择时加入了跳跃特征和状态转移特征**

**3.通过每次删除一个特征来判断单个特征的重要性，从而实现特征的取舍，降低训练难度**

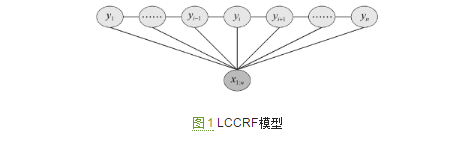
**4.在实验阶段，对梯度计算进行了并行化**

**5.在评估阶段，对封闭集和开放集都进行评估并交叉验证**

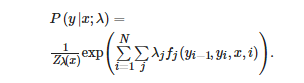
**6.“连续S”型错误是指分词程序将原本应标记为 {B，E}或{B，M，E} 的连续单字标记为 {S，S} 或 {S，S，S}。 这种错误常常由文本中的专业词汇导致，可以利用额外的辅助词典和前后向匹配算法进行修正。**

中文分词是信息检索工作的一项先决任务。随着大数据时代的到来, 信息检索工作对于中文分词准确率和召回率的要求也不断提高。该文提出了一种针对中文短文本的分词方法。该方法首先利用机器学习中的条件随机场模型对待处理的中文短文本进行初步分词, 然后再利用传统词典分词方法对初步分词结果进行修正, 从而完成分词工作。针对中文短文本的特点, 该方法在条件随机场的标记选择和特征模板编写上做了相应优化

CRF模型是机器学习领域一种用于序列标记的无向图模型。对于一个任意的CRF模型，一方面其数学性质难以用简单的数学表达式进行描述，另一方面其训练和推测工作还没有非常有效的算法。因此，在序列标记和中文分词的研究中，本文采用CRF模型的一种特殊形式——线性链CRF(linear chain conditional random field，LCCRF)模型，如图1所示。



LCCRF模型从数学上看对应一个有限状态自动机，它的数学性质非常适合用于序列标记和中文分词的相关工作且其训练过程的时间复杂度也能够被控制在可以接受的范围之内。根据文[[6](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-6)]的随机场理论，在给定LCCRF模型和观测序列x的条件下，一个状态序列y出现的条件概率为



其中：特征函数*fj*(*yi*-1,*yi*,*x,i*)是一个正实值函数，描述在给定*x*的条件下，某一*y*在位置*i*上状态转移的信息。*j*代表特征函数的编号，*λ*j是第*j*个特征函数的系数，表征特征函数的权重，也是LCCRF训练过程中需要优化的主要参数。*Zλ*(*x*)是归一化因子，它使得所有合法的*y*出现的条件概率之和为1：



LCCRF模型在应用时分为训练和推断(测试)这2个阶段[[7](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-7), [8](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-8)]。推断过程中，对于给定的x，最有可能的*y*可以由Viterbi算法[[9](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-9)]计算得出



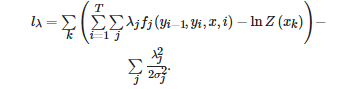
在训练过程中，LCCRF模型中参数值λ由极大似然估计方法计算得出。极大似然估计方法的描述如下：求出一组*λ*，使得训练集中所有*y*关于*x*和*λ*的条件概率联合乘积最大：



为简化计算，待优化的目标函数整理为对数形式：



实验所用的训练语料集将产生约106数量级的特征函数，常用优化算法的空间复杂度难以满足在一般处理机上实现训练过程的要求，因此在训练过程需要引入L-BFGS算法[[10](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-10)]来优化目标函数。另外，为了防止过拟合现象影响LCCRF模型的泛化能力，本文在目标函数中引入惩罚因子，以减小较大的λ对目标函数的影响。式(1)经过化简并引入惩罚因子后，最终整理为



最后，利用L-BFGS算法优化，求得最优的λ即可完成训练过程

相比自然语言处理领域的其他机器学习模型，LCCRF 模型在序列标记问题的处理中有3个较为显著的优势： 1) LCCRF 模型允许任意的特征选取和长程的依赖关系，因此它能够更加精确地描述实际问题和更加准确地进行标记工作;2) LCCRF模型在计算某一状态序列的条件概率时采用全局的归一化因子，这使得它能够规避MEMM中的标记偏置问题;3) 在LCCRF模型训练过程中，待优化的目标函数是一个凸函数，这保证了训练过程可以在有限时间内收敛并收敛至全局最优值。

**2 利用条件随机场模型实现中文分词**

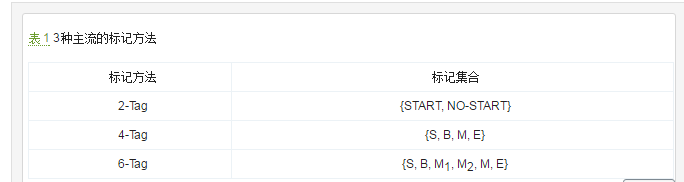
**2.1 中文短文本分词的实现方法**

针对短文本中文分词任务可以被抽象为序列标记任务。首先，给定若干特定标记，用于标记待分词的中文短文本中的单字。每种标记表征它所标记的中文单字在分词结果中所在的位置。然后，根据中文的语法规则(即中文单字间的关联性)和实验数据，来确定对于当前语料最优的特征模板。并根据标记方法、特征模板和已有的语料训练一个用于分词的LCCRF模型。最后，利用训练完毕的LCCRF模型对待分词的中文短文本做相应的分词标记，从而获得分词结果。

本文提出的分词方法主要针对在信息检索任务中经常遇到中文短文本。这类文本一般由有十几至几百个汉字组成，较短的文本长度决定了文本中语义信息的相关度较大。其中所含词汇一般是出现频率较高的常见词汇或者是具有一定特殊性的专业词汇。因此，在标记选择和特征选择上，该方法将针对短文本的这些特性，对于一般的LCCRF分词方法做出优化和改进。

**2.2 标记选择**

目前有3种主流的标记方法即2-Tag、4-Tag和6-Tag方法[[11](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-11)]，如表1所示。



实验数据表明，对于一个基于LCCRF模型的分词系统，采用6-Tag 标记方法进行训练能够使分词系统获得最高的准确率和召回率。 然而统计数据表明：中文文本特别是中文短文本中由5个(含)以上汉字所组成的词汇所占比例极小，常见短文本中的词汇是由4个(含)以下的汉字构成。6-Tag 方法中的 M2 标记难以起到它应有的作用，还额外产生了很多不必要特征函数，这些特征函数将影响LCCRF训练过程所需的时间和内存空间。 因此，本文提出了一种新的标记方法——5-Tag 标记方法。5-Tag 标记方法既可以完整保存汉语中所有4字(含)以下词汇所蕴含的关联信息，又最大限度地避免产生了不必要的特征函数，因此显著降低了LCCRF模型训练过程的时间和空间复杂度。在5-Tag 标记方法中，单字词被标记为S，多字词的起始位置和结束位置上单字会分别被标记为B和E，其他位置上单字会依据词汇中单字数目不同被标记为I或M，如表2所示。



在Sighan bakeoff 2005的PKU语料集上的测试结果如表3所示。可以看出：5-Tag方法在取得了与6-Tag方法大体相当的准确率和召回率的同时，大幅减少了特征函数的数量。随着特征函数数量的减少，训练过程所用的时间和训练算法所消耗的内存都随之下降。因此，可以认为在针对中文短文本这一特定对象时，5-Tag方法优于6-Tag方法。另外需要指出的是，在标记对比的实验中，为了提高3种标记方法的区分度，在特征选择时没有直接选择表4的模板，而是在表4的模板的基础上，舍弃了其中的三元特征。





**2.3 特征选择**

训练和使用分词系统前需要为分词系统编写特征模板，LCCRF模型的训练算法会根据特征模板生成相应的特征函数，因而特征模板的选择会直接影响LCCRF模型及分词系统的性能。因此，选择合适的特征模板是提高分词系统性能的另一项重要工作。针对中文短文本的特性，本文在设计模板时采用了“常用特征结合针对特定环境的补充”的方法。表4 给出了本次实验中采用的特征模板。尽管在很多已有的基于LCCRF模型分词算法中，词性被视作一个关键特征，但是语料中词性标注的工作需要人工来完成，因此本文提出的系统暂不考虑引入。值得注意的是，除了常用的一元特征和二元特征外[[12](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-12)]，本文引入了状态转移特征，用以描述不同标记间固有的转移关系。例如，标记E之后的标记只可能是标记S或B，而标记B之后有很大概率出现标记I或E。

由于特征模板的编写需要同时引入多个特征，这些特征形成的联合作用对分词结果产生影响，因此很难准确评估单一特征的影响。表5给出了一组实验数据，说明了表4的模板在当前语料集及实验环境下具有相对较好的性能。



从表5中可以看出，在移除二元特征后，分词系统的F-Score值有了显著的下降，因而在当前实验环境下，二元特征对于实验结果有较为明显的影响，而三元特征和跳跃特征对于实验结果的影响则相对较小。值得注意的是，在添加了不合适的特征后，不但系统的F-Score指标有所下降，而且其训练过程的时间和空间复杂度因为特征数量的增加而随之上升。

**3 实验和评估**

**3.1 实验设计**

Sighan bakeoff 2005语料集被广泛应用于中文分词实验和相关测试。因而在本次实验中，本文选择Sighan bakeoff 2005的4个语料集作为训练和测试数据集进行实验。每一个语料集将被随机分为2个部分，其中80%的数据作为模型训练数据和封闭性测试的测试数据，另外20%的数据作为开放性测试的测试数据。同时，实验中使用交叉验证的方式来帮助选择相关参数、标记方法和特征函数。实验平台为一台配备Intel Xeon E5-2420 1.9GHz (6核心，12线程) CPU和8 GB内存的服务器。

**3.2 梯度计算并行化**

LCCRF模型的训练时间要远远大于同等规模的HMM和MEMM，模型训练时间成为本文方法的主要开销。为了将训练时间控制在可以接受的范围之内，加速机制被引入到训练过程中。

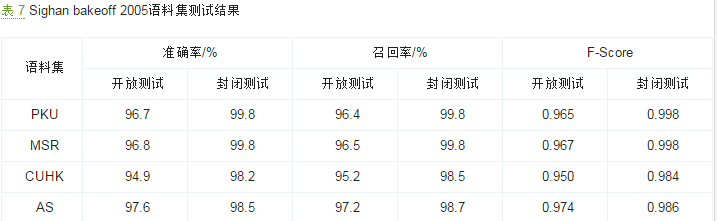
一个可以有效减少训练时间的方法是采用并行算法来计算目标函数的梯度值。因为目标函数的梯度值实质上是每一组训练数据产生的梯度值分量之和，所以很容易将梯度计算任务分解为可以在多个线程中同时执行的并行计算任务。每个线程中运行的任务计算特定的若干组训练数据产生的梯度，最后由一个线程将各线程的计算结果相加即可。不同并行度下的训练算法的时间开销如表6所示。实验证明：在单处理机上最佳的并行度与当前处理机上CPU所支持的最大线程数相同。



另一种并行加速机制是将目标函数的梯度计算分配到多个处理机上进行。从理论上讲，多处理机并行算法能够取得比简单的多线程并行算法更快的速度。但是由于实验条件的限制，本文在这里不涉及多处理机算法的研究。

**3.3 在Sighan bakeoff 2005 语料集上的实验结果**

在本次实验中，本文提出的系统在每个语料集上都将被分别进行开放测试和封闭测试。在封闭测试中，测试数据与训练数据完全相同，而在开放测试中测试数据与训练数据来自不同某一语料集的不同部分[[13](http://jst.tsinghuajournals.com/CN/rhhtml/20150816.htm" \l "R-13)]。本文提出的分词系统在 Sighan bakeoff 2005 四个语料集上的测试结果如表7所示。可以看出：本文提出的分词系统在Sighan bakeoff 2005的4个语料集的开放测试中均取得了0.950以上的F-Score，特别是在PKU、MSR和AS这3个语料集中均达到0.965以上，分词系统的准确率和召回率已经基本达到了实际应用的标准。



**3.4 辅助词典和修正算法**

通过分析和对比分词结果，发现在所有4个语料集的分词结果中，有一种被称为“连续S”型错误的出现频率较高。“连续S”型错误是指分词程序将原本应标记为 {B，E}或{B，M，E} 的连续单字标记为 {S，S} 或 {S，S，S}。 这种错误常常由文本中的专业词汇导致，可以利用额外的辅助词典和前后向匹配算法进行修正。

修正算法首先扫描初步分词结果，如果在扫描中发现有连续的标记S，则检查这些连续S所对应的中文单字是否能够组成词汇或词汇前缀，然后根据检查结果和后续中文单字的标记做出相应处理。需要指出的是，这种修正虽然有可能导致原本正确的标记变为错误的标记，而且会降低召回率，但是这种方法仍然有效提升了分词方法的准确率。

**4 结 论**

本文提出了一个基于LCCRF模型，结合辅助词典的针对短文本的中文分词方法。该方法首先利用LCCRF模型对待处理的中文短文本进行初步分词，再用传统匹配算法和辅助词典对初步分词结果进行适当修正，取得了较为理想的准确率和召回率。该系统在Sighan bakeoff 2005的PKU、MSR、CUHK和AS等4个语料集上的开放测试分别取得了0.965、0.967、0.950 和 0.974 的F-Score。下一步将在训练算法并行化、更优特征模板的引入和特定分词任务的专门优化等方面进行改进。