基于 CRFs 的中文分词和短文本分类技术

Chinese Word Segmentation and Short Text Classification Techniques Based on CRFs

（申请清华大学工学硕士学位论文）

培 养 单 位 ： 计算机科学与技术系

学　　科 ： 计算机科学与技术

研 究 生 ： 滕 少 华

指 导 教 师 ： 孙 茂 松 教 授

二〇〇九年五月

基于CRFs的中文分词和短文本分类技术

滕少华

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后遵守此规定）

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘 要

中文自动分词和短文本分类是自然语言处理中的基础任务，本文将介绍关于这两个领域的若干基于条件随机场(CRFs)的算法以及在此基础上的创新。

本文的工作主要包括两部分：第一部分，我们借用了文本分类领域的特征选择算法对中文分词中的特征进行分析。分析结果表明，特征选择算法在中文分词的任务中也是适用的。在中文分词领域，基于字标注的方法得到广泛应用。通过字标注系统，中文分词任务被转换为序列标注任务，许多成熟的机器学习算法得以应用。评测结果表明，在众多的机器学习算法中，基于 CRFs 的分词器可以达到 state-of-the-art 的分词效果。CRFs 分词器对于给出的每一个切分，都可以提供置信度。在本文中，我们深入调研了 CRFs 提供的置信度，在此基础上，提供了一种基于置信度的后处理中文分词算法。三个不同数据集上的实验结果证明，我们的算法是有效的。另一方面，我们对未登录词(OOV)在上下文中的分布进行观察，提出了一种基于篇章内部信息和 CRFs 置信度的 OOV 识别方法，可以进一步提高中文分词的准确度。

第二部分，我们借用了中文文本分类中的字标注算法来解决短文本分类问题。通过标注算法，可以将短文本分类问题转化为序列标注问题，这样 CRFs 就可以用于短文本分类任务中。实验结果表明，基于 CRFs 的短文本分类器可以达到更高的分类精度。

关键词：中文分词　短文本分类　条件随机场　特征选择　置信度　未登录词

Abstract

Chinese word segmentation (CWS) and short text classification (STC) are both basic tasks in natural language process (NLP). In this paper, some novel methods based on CRFs are introduced for these two NLP tasks.

First, in this paper, we borrow the idea of feature selection from text classification to evaluate each feature's contribution in CWS task. Our analysis demonstrates that feature selection methods are useful in CWS task. In CWS task, the most widely used methods are character-based tagging method, which reformulates CWS task to a sequence tagging task. It is demonstrated by previous work that CRFs tagger can achieve state-of-the-art performance. Given a word segmentation proposed by the CRFs, we can compute a confidence in each segment. In this work, we investigate the confidence generated by CRFs and propose a novel post-process method to improve the CWS performance. We conduct experiments on three corpora which show our CRFs confidence approach achieves better performance. On the other hand, we analyze OOVs distributions in context. Based on our analysis, we propose a method to use in local information to recognize OOVs.

Second, we borrow the character-tagging method in CWS task to solve STC problem. After converting the classification problem to a sequence labeling problem, CRFs can be used in the STC task. Experiment results show that CRFs based classifier can produce a promising performance in STC task.

Keywords: Chinese Word Segmentation Short Text Classification Conditional Random Fields Feature Selection Confidence OOV

目 录

第1章 引言

1.1　课题背景及意义

随着信息化技术的迅猛发展，互联网上的信息量呈现指数爆炸型增长趋势，如何更好的管理和组织这些信息已经成为了热门的研究方向。在这些日益增长的信息中，大部分信息是以半结构化文本或纯文本的形式出现的。如今，人们所面临的不在是信息缺失的时代，而恰恰相反，摆在我们面前的海量文本信息使得文本信息的挖掘成为迫切的需求。单纯地依靠人工的处理已经远远根本上信息的产生速度，越来越多的研究者正着手于利用计算机进行快速、准确的文本信息处理。与英语等西方语言不同，中文文本中并不存在关于“词”这个单位的天然分隔符，中文自动分词是中文信息处理的基础步骤。中文分词技术[1, 2 ,3](Chinese Word Segmentation, 以下简称 CWS)和文本自动分类技术[4, 5, 6, 7](Text Classification, 以下简称 TC)的产生，在一定程度上解决了文本信息组织的难题。

图 1.1　中文分词技术在搜索引擎中的应用

随着搜索引擎技术的广泛应用，中文分词和作为一种特殊的文本分类技术的短文本分类技术(Short Text Classification, 以下简称 STC)得到重视。对于搜索引擎来说，最重要的并不是找到所有结果，因为在上百亿的网页中找到所有结果没有太多的意义，最重要的是把最相关的结果排在最前面，这也称为相关度

排序。中文分词的准确与否，常常直接影响到对搜索结果的相关度排序。另一方面，短文本分类技术作为一种特殊的文本分类，在 Web 环境中得到了广泛的应用，很多任务，如 query 分类，文章标题分类等，都需要使用短文本分类技术。

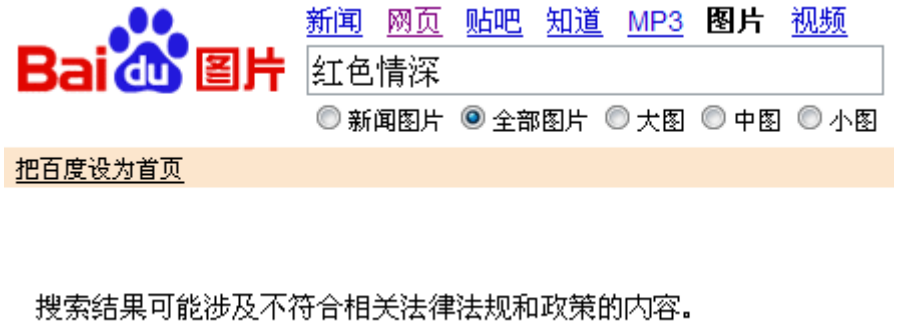

图 1.2　短文本分类技术在搜索引擎中的应用(query 过滤)

1.1.1　中文分词问题

作为大规模文本处理的基本功能，中文自动分词技术的产生，在一定程度上解决了文本信息处理的难题。近些年来，利用计算机的强大功能对中文文本进行自动快速的分词引起了研究人员的重视。在过去的十年中，中文分词领域得到了迅猛的发展。中文分词领域的主流方法，由传统的基于词典和规则[8, 9]的方法，向基于统计的方案转变[10, 11, 12]。统计方案中，基于字标注的分词法[3]，基于子串标注的分词法[13]层出不穷。给定人工标注好的训练集文档，通过选择合适的机器学习方法(例如：支持向量机 Support Vector Machine(SVM) [14]、最大熵 Maximum Entropy(ME)[15]、最大熵马尔可夫模型 Maximum Entropy Markov Model(MEMM)、条件随机场 Conditional Random Fields(CRFs)[16]、感知机 Perceptron [17])，我们可以得到训练好的模型，将这个模型应用于未切分的测试集样本，完成整个自动切分的过程。同时从这里我们也不难发现，中文自动分词作为自然语言处理的经典问题，和模式识别[18]、机器学习[19, 20]、信息检索[21, 22]等相关领域有着非常密切的联系。

经过近些年的发展，尤其是 2003 年国际中文分词评测活动 Bakeoff 开展以来，有了统一的训练与测试语料，回避了“词”的定义这样一个棘手的问题。通过“分词规范+词表+分词语料库”的方法，使词语在真实文本中得到了可计算的定义，这是实现计算机自动分词和可比评测的基础。随着分词算法的不断

改进，目前在各个已知的封闭的公共数据集上，最好的分词器的性能已经相差无几。目前的评测结果表明，基于 CRFs 的分词器可以达到 state-of-art 的分词效果，所以本文中的研究工作主要建立于 CRFs 分词器基础上，但是，面向开放语料的分词性能仍然不能令人满意。随着 Web 上的中文语料爆炸性的增长和对中文语料自动分词的需求不断加强，面向 Web 的中文自动分词技术逐渐引起了人们的重视。和传统的中文自动分词不同的是，Web 环境下的分词面临着更为复杂的问题：

1. Out-of-Vocabulary(OOV)的问题：由于 Web 上的文本变化较为剧烈，每时每刻都有新的文本产生，旧的文本消亡。预先训练好的分类模型可能并不适应于 Web 的环境。训练语料只能覆盖有限的一部分词语，加之新的词汇不断地在 Web 文本中涌现，OOV 在面向 Web 的中文分词处理中成为非常严峻的一个问题。
2. 分词系统的速度：每天产生大量的待分词的中文语料需要及时处理，高速的分词系统成为迫切要求，如何开发出可以适应于 Web 环境下的快速分词系统值得我们进行研究。

1.1.2 短文本分类问题

通过文本分类，人们可以将文本分门别类的进行组织、索引和查找等。传统的文本分类技术通常应用于图书馆的图书分类、专利局的专利文献分类等，这种方法的优点在于分类精度较高，其主要缺陷在于需要人工进行干预，且需要较多的背景专业知识，处理速度慢，无法应用于海量文本的处理。近些年来，利用计算机的强大功能对文本进行自动快速的分类引起了研究人员的重视。给定预先定义好的分类体系，以及人工标注好的训练集文档，通过选择合适的机器学习方法(例如：支持向量机 SVM [14， 23]、朴素贝叶斯 Naïve Bayesian [5]、k-近邻 KNN [6])，我们可以得到训练好的模型，将这个模型应用于未标注的测试集样本，完成整个自动分类的过程。 短文本分类问题作为文本分类问题的一个分支，除具有共性之外，还面临一些特殊问题需要解决，单纯地从普通文本分类任务中移植的算法有时并不能得到很好的效果，其主要问题在于：

1. 文本特征的稀疏性，短文本的长度相对于普通文本要短很多，因此每篇文章中的特征很少，故传统的文本分类特征选取方法在该任务中并不适合。

2. 短文本有其特殊的性质，传统的分类算法并没有考虑这些因素。

1.1.3　研究目标

本文拟对上述问题进行初步的探讨和研究，着眼于提出一套可行的解决方案，并在一定的程度上解决上述问题。

1. 引入特征选择方法，对中文自动分词所使用的特征进行分析和评价，从我们的分析中可以得到如下结论：使用的特征中，只有很少一部分起到了很重要的作用，另外很大一部分特征在分词的过程中几乎没有作用。通过特征选择选取有意义的特征，可以有效降低分词器的时间与空间开销。
2. 利用 CRFs 分词器提供的置信度，对分词结果进行后处理，进一步提高分词精度。
3. 针对 OOV 问题，利用文本篇章中的统计信息来辅助分词，将新词发现和中文分词的任务结合起来，可以得到较好的分词效果。
4. 利用中文分词中的字标注方法，将短文本分类问题转化成序列标注问题，从而可以使用 CRFs 完成该任务，在不使用外界资源的情况下，提高短文本分类的精度。

1.2　中文自动分词研究现状

与英语等西方语言不同，中文文本中并不存在关于“词”这个单位的天然分隔符，中文自动分词是中文信息处理的基础步骤。中文分词，顾名思义，是将自然语言表示的中文文本，通过一定的技术手段，根据其内容，将其词与词之间用预先定义好的分隔符隔开，成为切分好的文本。传统的分词方法是基于词典和规则的，如前向最大匹配 (FMM), 反向最大匹配 (BMM)等。在此基础上，基于统计的分词方法逐渐兴起。尤其是 SIGHAN 中文分词评测活动开展以来，中文分词技术得到了长足的发展。SIGHAN 提供了标准的训练和测试语料，避免了“词”的定义这个棘手的问题，专注于改善分词算法。因为使用的是标准的训练集和测试集，不同的方法之间可以互相比较，另一方面，也方便了分词系统的改进。

从 SIGHAN2003，SIGHAN2005，SIGHAN2006[24, 25, 26]上的结果可以分

析得到，将分词问题转化为序列标注问题已成为主流方法，因为通过这一转化，可以引入现有的机器学习方法进行分词，现有的机器学习方法建立在鲁棒的统计推断基础上，可以得到比传统的“词典+规则”的分词方案更高的性能，在中文分词领域中，基于有监督的、无监督的和混合的模型都有所应用。目前主流的中文分词方案如下：首先，通过引入标注系统，将原有的中文分词问题转化为序列标注问题或是分类问题，接下来，利用事先手工标注好的训练集文本集合，结合合适的机器学习方法(CRFs 等)，得到训练好的模型。在测试步骤中，将训练好的模型应用于未标注的测试集样本，并和正确答案进行比较，得到最终的切分准确率评价，从而完成了整个中文自动分词的过程。目前的评测结果表明，基于 CRFs 的分词器可以达到 state-of-art 的分词效果。

在下面的子章节中，我们将介绍典型的中文自动分词系统的大致步骤以及目前的一些研究现状。

1.2.1　数据集

由于中文分词任务的特殊性，标准数据集占有非常重要的地位。因为“词”的精确定义还是一个尚未解决的问题。汉语语法教科书对“词”的定义是相当抽象的：语言中有意义的能单说或用来造句的最小单位。在计算上，这种模棱两可的定义是不可操作的，或者说，是不可计算的。即使在母语为汉语的话者之间，对中文词语的平均认同率只有 0.76 左右[27]。因而，标准数据集的存在，可以使研究者不去深究词的定义，而集中精力于改进算法，如何从训练语料中挖掘到尽可能多的有意义的知识。允许在词的定义上存在争议，但是同源的训练与测试语料中所遵循的分词标注是一致的。在我们的工作中，使用的数据集包括 SIGHAN2005 的 PKU 和 MSR 数据集，以及 YUWEI 语料库。

1.2.2　转化为序列标注问题

将原有的切分问题转化为序列标注问题是重要步骤，通过使用标注系统来完成这一步骤。在众多的标注方法中，LMRS，也称为 IOB，BIES 标注系统得到了最广泛的应用。下面我们以常见的字标注方法为例，介绍该标注系统的具体标注方法。在 LRMS 标注系统下，每个字依据其在词中出现的位置，给与不同标签。L(left)代表词的左边界，R(right)代表词的右边界，M(middle)代表词的中间部分，S(single)代表单字成词。如下面这个例子：

我/爱/北京/天安门/。

S　S　LR　L M R S

使用了这样的标注系统，我们就可以把中文分词问题转化为序列标注问题。对于一个未分词的句子，如果可以对每个字给出标签，那么就等效知道了分词结果。通过这样的转化，很多机器学习方法可以在分词领域中得到应用。另外，由于汉语本身的性质，有的字倾向于出现在词的固定位置，如“我们”的“们”，一般出现在词的右边界，而“生”，则可能出现在词中的位置是不定的，如“花生”，“生产”等例子。通过将中文分词问题转化为序列标注问题，可以利用相对固定的字来推断相对不定的字的位置信息。同时，我们也可以把转换后的序列标注看成一个分类问题，句子中的每个位置被分为 LRMS 四类。分类问题中有很多特征选择方法，通过这样的标注，我们可以借用特征选择方法对中文分词中所用到的特征进行评价，使用最有效的特征，降低特征数目，从而提高分类器的效率。我们将在第二章中详细阐述这一问题。

1.2.3　条件随机场算法

目前的评测结果表明，基于 CRFs 的分词器可以达到 state-of-art 的分词效果。CRFs 在中文分词领域得到了广泛的应用。CRFs 是无向图模型的一种形式,它采用了链式无向图结构计算给定观察值条件下输出状态的条件概率。在给定观察序列的情况下，标记序列的条件概率为:

$$p_\theta(y \mid x) \propto \exp\Big(\sum_{e \in E,k} \lambda_k f_k(e, y|_e, x) + \sum_{v \in V,k} \mu_k g_k(v, y|_v, x)\Big) \quad (1\text{-}1)$$

其中，$f_k(e, y|_e, x)$ 为状态转移特征函数，$g_k(v, y|_v, x)$ 为状态特征函数，λ_k，μ_k 是由训练样本得到的特征函数权重。计算特征权重函数采用极大似然估计方法。CRFs 指数模型为凸函数，可以采用迭代方法找到全局最优解。目前常用的是 L-BFGS 迭代方法。

在中文分词任务上的表现，CRFs 要优于隐马尔可夫模型(Hidden Markov Model，简称 HMM) 和最大熵马尔可夫模型(MEMM)。HMM 是一种产生式模型，它有着比较强的独立性假设，导致其不考虑上下文的特征，限制了特征的选择. 而 CRFs 则可以任意选择特征。另一方面，产生式模型需要对联合分布做估计，在中文分词任务上的性能要低于判别式模型。

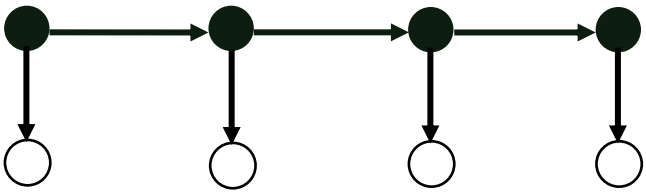

图 1.3　隐马尔可夫模型（黑色代表隐藏节点，白色代表观察到的节点）

MEMM 是对 HMM 的一种改进,它克服了生成属性所出现的问题以及 HMM 的严格的独立假设。MEMM 是一种判别式模型，只需要对条件分布做估计。但是,它却存在一个缺点就是标记偏见问题(label bias problem)，产生这种问题的原因就在于 MEMM 对于状态序列的计算采用的是局部归一化的方法。而 CRFs 采用的是全局归一化的方法，从而克服了标记偏见问题。

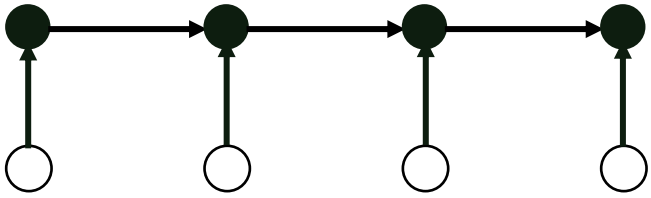

图 1.4　最大熵马尔可夫模型（黑色代表隐藏节点，白色代表观察到的节点）

CRFs 是一种判别式模型，采用的是无向图分布，没有严格的独立性假设，可以任意选取特征，而且因为采用了全局归一化的方法，避免产生标记偏见问题，所以在中文分词任务上取得了比较好的效果。其中，链式 CRFs 在中文分词任务中最为常用，本文的后续工作即在此基础之上展开。

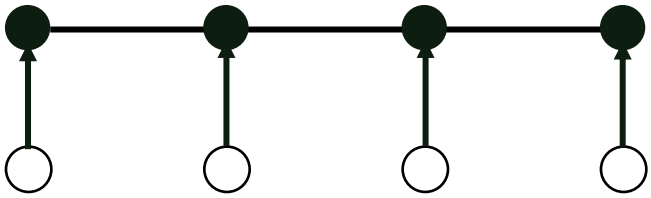

图 1.5　链式条件随机场模型（黑色代表隐藏节点，白色代表观察到的节点）

1.2.4 性能评价

中文分词常见的性能评价指标有准确率 Precision、召回率 Recall 和 F1-Measure。对于某个切分结果，准确率 Precision 指的是给出的切分中正确的比例，而召回率 Recall 指的是给出的切分占实际正确切分的比例。这两个指标都来源于信息检索，是自然语言处理任务中较为常用的指标。

通常来说，准确率 Precision 和召回率 Recall 呈轻微的负相关关系，为了更好的评价系统的指标，人们定义了二者的调和平均作为综合评价指标，即 F1-Measure，其具体的定义为：

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \qquad (1-2)$$

此外，中文文本分类任务中还有一些较特殊的评价指标，如 IV (in-vocabulary words) recall，OOV (out-of-word vocabulary words) recall 等，分别用来评价词典词的召回率和未登录词的召回率，一般来说，这两者之间也呈轻微的负相关关系。

1.3 短文本分类研究现状

短文本分类，作为一种特殊情况下的文本分类，与文本分类任务有很多共同点。其大致流程是将自然语言表示的短文本，通过一定的技术手段，根据其内容，将其划分到预先定义好的一个或者多个类别中。通常情况下，我们会利用事先手工标注好的训练集，结合合适的机器学习方法，得到分类器模型。在测试步骤中，将训练好的分类器模型应用于未标注的测试集样本，并和正确答案进行比较，得到最终的分类准确率评价，从而完成了整个短文本分类的过程。

与传统的文本分类问题类似，短文本分类问题在不同情况下可以细分为以下几种类别。当集合中的文本属于且仅属于一个类别时，该文本分类任务成为单标签(single-labeled)的文本分类问题；相反，当文本可以属于一个或者多个类别时，则对应的称作多标签(multi-labeled)的文本分类问题。

此外，还可以根据待分的类别数对任务进行分类：当只有两个类别时，称为二分类问题(binary classification)，常见的有垃圾邮件判定、主题网页抓取等。而更为常见的是多类别分类问题，例如：门户网站关于新闻的分类经常划分为国际、国内、娱乐、财经、体育等多个类别。

在下面的子章节中，我们将介绍典型的短文本分类系统的大致步骤以及目前的一些研究现状。

1.3.1　数据预处理和常用数据集

这部分的工作主要包括：

1. 收集训练和测试用的短文本集合，对训练测试集合进行合适的划分(通常情况下训练集文本要多余测试集文本为 7:3，70%文本作为训练，剩下的30%作为测试评价算法效果)，短文本的来源很广泛，如查询日志，文章标题，用户反馈等。
2. 去除数据集中的噪音，包括乱码、非文本内容等。与传统的文本分类相比，短文本分类数据的噪声问题更为严重，因为在短文本的条件下，每篇文章中的特征数较少，噪声会对最后的分类结果产生严重的影响。我们必须要将更多的精力放在去除噪音数据上，网页中存在较多的结构化或半结构化信息，广告信息等，这些对于分类任务来说都是需要过滤的。
3. 针对不同语种的文本内容进行特殊处理，对于英文文本，我们一般要对其进行去除停用词(stop-words)、保留词干(stemming)等。对于中文文本，由于其并不存在天然的空格来区分词和词之间的距离，我们必须要对其进行分词或者 N-Gram 的切分。

传统文本分类研究中会使用到通用的标准的数据集合。常见的英文数据集有 Reuters-21578，OHSUMED 和 20 Newsgroups。其中，OHSUMED OHSUMED 收录了 1987-1991 年 Medline 医药记录，由 Hersh 于 1994 年整理。为了用于文本分类，Lewis 和 Yang 对原文本集进行了加工，生成的文本集包含 233445 篇文本，每篇文本只含有标题和摘要，共有 119 类，其中 1987-1990 年的文本作为训练集，1991 年的文本用作测试集。

在中文方面，目前暂时没有标准的评价数据集。本文中采用的数据集搜狗实验室公布的网页标题集合，共分 15 类，包括 429819 篇文档。训练集和测试集未划分。

1.3.2　文本向量化处理

与传统的文本分类任务类似，为了让计算机能理解自然语言表示的文本内容，我们采取一定的措施将短文本进行向量化处理。其中主要包含以下几个步骤：

1. 选择合适的特征单元，对文本进行标引(document indexing)。文档索引在信息检索等其他领域也有着举足轻重的地位。而在文本分类中，通常采用的是词袋模型(Bag-of-words Model, BOW)，其假设每个特征单元[26](可以是词，也可以是 N-Gram，甚至短语级别)对应特征空间中的一维，这样文本就可以表示为特征空间中的词频向量。
2. 权重计算，直观上来说，每个特征单元在文章中的重要性有所差别。一般来说，频度越高的特征单元权重越大。同时，我们也需要考虑停用词带来的噪音影响。通常来说，最为常用的权重计算方法为 tf*idf 权重计算方法[28]，如无特殊说明，本文采用其作为默认的权重计算方法。
3. 特征选择，由于自然语言的特殊性，在传统的文本分类任务中，用于文档标引的特征空间一般具有较高的维度，这将大大提高在训练分类器模型时的训练开销，另外，其中包含的一些噪音特征对最终分类器的性能也有不利的影响。因此，我们有必要对原始空间中的特征进行选择，得到一个更为有效的特征子空间。但是，在短文本分类任务中，特征稀疏的问题非常严重，在进行特征选择时要非常谨慎。

通过上述的几个步骤，我们将文本表示成计算机所能理解的形式。需要特别指出的是，对于训练集和测试集中的文本，我们必须要用同样的方式进行处理，即：特征单元、特征选择方法和权重计算都必须相同，这样才能在同样的分类器模型下进行学习和测试。

1.3.3　分类器选择

经过近些年的发展，已经有若干机器学习方法应用在文本分类问题上。文本分类问题可以看作是新的方法的测试和评价平台。较为常用的分类方法有：朴素贝叶斯分类器、Rocchio 分类器、k-近邻分类器、人工神经网络分类器和支持向量机分类器等。目前，支持向量机分类器可以达到 state-of-art 的性能。

1.3.4　性能评价

短文本分类任务和传统文本分类任务的评价指标相近，关于文本分类的性能评价，目前有多种指标[29]。

此外，针对多类别分类问题，有两种综合的评价指标：

一种是微平均方法，即将所有类别的混淆矩阵合并在一起求总体的指标，这个评价方法将所有的文本都看成是同等重要；

另一种是宏平均方法，即对每个类别的混淆矩阵分别求各自的指标，然后再求的平均数得到最终的结果，这个评价方法将所有的类别都看成是同样重要，相对微平均而言，宏平均对于小类别的文本比较敏感。一般来说，小类别的分类效果不如大类别(通常情况下，训练集文本数越多则训练效果越好)，所以宏平均值通常小于微平均值。

在文本分类任务中，微平均更多的作为评价系统的指标。在本文中如无特殊说明，均采用微平均的计算方式。另外，不难证明，当所有的文本属于且仅属于一个类别时，在微平均条件下，所有类别的准确率、召回率和 F1 具有相同的值。本文中的实验即属于这一情况。

1.3.5　短文本分类任务的特殊性

短文本分类问题作为文本分类问题的一个分支，除具有共性之外，还面临一些特殊问题需要解决，因为文本长度短，特征稀疏，难以衡量短文本之间的相似性，单纯地从普通文本分类任务中移植的算法有时并不能得到很好的效果。但是，短文本分类任务在 web 环境下得到了越来越广泛的应用。如何有效地提高短文本分类准确度，已成为一个至关重要的问题。

由于短文本分类任务的特殊性，传统的文本分类方法在该领域表现不佳。因为文本较短，面临特征稀疏的问题，文本之间很少含有相同的特征，这样，文本之间的相似性不好度量，这些特殊的性质为短文本分类任务增添了极大的困难。

前人的工作中，很多方法用来克服短文本分类中的特征稀疏问题。这些方法使用外部资源来增加文本之间共享的特征。外部资源的引入可以提高短文本分类的准确性。但是，外部资源的使用，比如说利用搜索引擎返回的结果，是非常耗时的步骤，所以，这些方法在对时间要求比较高的场合下，比如说实时系统中，是不合适的。

为了提高分类系统的速度，本章的工作中并没有使用外部资源，而是利用

短文本自身的特性来提高分类准确率。因为短文本的长度有限，所以主题集中，短文本的特征之间具有较强的一致性。该性质如果利用得当，可以有效地提高短文本分类的准确率。

1.4　本文的研究重点和内容安排

根据以上的综述，基于 CRFs 的分词器可以达到 state-of-art 的分词效果，所以本文中的研究工作主要建立于 CRFs 分词器基础上，但是，面向开放语料的分词性能仍然不能令人满意。随着 Web 上的中文语料爆炸性的增长和对中文语料自动分词的需求不断加强，面向 Web 的中文自动分词技术逐渐引起了人们的重视。和传统的中文自动分词不同的是，Web 环境下的分词面临着更为复杂的问题：首先，是关于未登录词的问题：由于 Web 上的文本变化较为剧烈，每时每刻都有新的文本产生，旧的文本消亡。预先训练好的分类模型可能并不适应于 Web 的环境。训练语料只能覆盖有限的一部分词语，加之新的词汇不断地在 Web 文本中涌现，OOV 在面向 Web 的中文分词处理中成为非常严峻的一个问题。其次，分词系统的速度问题：CRFs 是无向图模型的一种形式,它采用了无向图结构计算给定观察值条件下输出状态的条件概率，模型本身决定其时间复杂度和空间复杂度非常高，如何提高基于 CRFs 的分词器效率值得我们进行研究。

短文本分类，是一种特殊情况下的文本分类，但该任务具有特殊性，待处理的文本很短，每篇文章中的特征很少，故传统的文本分类特征选取方法在该任务中并不适合。本文的工作利用中文分词中的字标注方法，将短文本分类问题转化成序列标注问题，从而可以使用 CRFs 完成该任务，可以得到更高的分类准确度。

以上提到的内容将分别在本文的第二、三、四、五章中进行阐述，最后在第六章中我们将对本文的内容进行总结，并提出可能的改进和对未来的一些研究方向展望。

第2章 中文分词中的特征选择问题

2.1　字标注系统

将原有的切分问题转化为序列标注问题是重要步骤，通过使用标注系统来完成这一步骤。在第一章中，我们已经对 LMRS 字标注系统进行了简单介绍。在 LRMS 标注系统下，每个字依据其在词中出现的位置，给与不同标签。L(left)代表词的左边界，R(right)代表词的右边界，M(middle)代表词的中间部分，S(single)代表单字成词。

使用了这样的标注系统，我们就可以把中文分词问题转化为序列标注问题。对于一个未分词的句子，如果可以对每个字给出标签，那么就等效知道了分词结果。通过这样的转化，很多机器学习方法可以在分词领域中得到应用。另外，由于汉语本身的性质，有的字倾向于出现在词的固定位置，如“我们”的“们”，一般出现在词的右边界，而“生”，则可能出现在词中的位置是不定的，如“花生”，“生产”等例子。通过将中文分词问题转化为序列标注问题，可以利用相对固定的字来推断相对不定的字的位置信息。

这里我们注意到，通过字标注系统，我们将中文分词任务实际上转化成了一个分类任务，句子中的每个位置被分为 LRMS 四类，如果每个位置的类别标签知道了，那么，也就等效于句子的切分方案知道了。分类问题中有很多特征选择方法，很自然的想法是借用已有的特征选择算法，对中文分词任务中所用到的特征进行评价，通过这样的标注，我们可以借用特征选择方法对中文分词中所用到的特征进行评价，使用最有效的特征，降低特征数目，从而有效地降低分词器的时间和空间复杂度。在下文中，我们将使用文本分类领域最常用的 $Chi^2_{\max}$ 特征选择算法。

2.2　$Chi^2_{\max}$ 特征选择算法

文本分类任务中经常要用到特征选择，因为不同的特征对分类过程的贡献是不同的。比如说一个简单的二分类任务，需要把文章分成两类：“体育”和“美食”。那么，特征“足球”将强烈地指向体育类，而特征“不错”则没有

这么大的区分度，因为它在两类文章中出现的概率相差不大。不同特征对分类过程的贡献不同，所以特征选择算法在文本分类任务中普遍应用。选择有效的特征，既可以提高分类系统的效率，同时也能避免不必要的噪声干扰。

在文献[30]中，作者对于特征选择方法进行了全面的分析和实验验证。结论表明$Chi^2_{\max}$特征选择方法在多个数据集上有较好的分类效果。实际上，$Chi^2_{\max}$统计量衡量了特征单元 t 和类别 c 的独立程度，我们首先介绍传统的$Chi^2_{\max}$统计公式如下。

根据特征单元 t 和类别 c 的统计信息，我们可以得到四个统计量。A 表示 t 和 c 同时出现的词数，B 表示 t 出现而 c 不出现的次数，C 表示 c 出现而 t 不出现的词数，最后，D 表示 t 和 c 都不出现的次数，N 表示训练集中的文本数，则开方统计量通过如下公式计算：

$$Chi^2(t,c)=\frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \quad (2-1)$$

理想情况下，t 和 c 同时出现或同时不出现，这表明 t 和 c 有很强的关联性。在这种情况下，如果某篇文档包含特征单元 t，那么我们很有理由认为该文档属于类别 c。另一方面，如果 t 和 c 完全独立，那么两者没有任何关联。对于特征单元 t，我们可以计算出其与所有类别的开方统计值。我们用其中最大的开方值代表 t 最终的开方值，其定义如下公式所示：

$$Chi^2_{\max}(t)=\max_{1\le i\le m} Chi^2(t,c_i) \quad (2-2)$$

$Chi^2_{\max}(t)$的大小即用来衡量该特征单元的重要程度。

2.3　中文分词任务中常用的特征

2.3.1　中文分词特征模板

中文分词领域常用的特征包括 unigram 特征，bigram 特征等，常用特征模板表示。以下面这个常用的特征模板为例，对其进行分析：

U00:%x[-2,0]

U01:%x[-1,0]

U02:%x[0,0]

U03:%x[1,0]

U04:%x[2,0]

U05:%x[-2,0]/%x[-1,0]

U06:%x[-1,0]/%x[0,0]

U07:%x[0,0]/%x[1,0]

U08:%x[1,0]/%x[2,0]

U09:%x[-1,0]/%x[1,0]

U10:%x[0,1]

其中，U01，U02 等指的是特征种类的序号，%x[0 ，0]指的是当前字(unigram)，%x[1, 0]指的是下一个字(unigram)，%x[-1, 0]/%x[1, 0]指的是前一个字和后一个字组成的二元组(bigram)。可以看出，特征模板中，每一行代表的是一类特征。下面我们举一个具体的例子：

我/爱/北京/天安门/。

S　S　LR　L M R S

假设我们采用这样的特征模板：

U01:%x[0,0]

U02:%x[-1,0]/%x[0,0]

假设当前观察的位置是“京”字，%x[0, 0]代表当前字，产生特征（“京”, R）;另外，%x[-1, 0]/%x[0, 0]指的是上一个字与当前字组成的二字串，这里产生特征(“北京”, R)。如果观察位置遍历了这个句子，那么就可以产生出这句话的所有特征：

(“我”， S)

(“爱”， S)

(“北”， L)

(“京”， R)

(“天”， L)

(“安”， M)

(“门”， R)

(“。”， S)

(“我爱”， S)

(“爱北”， L)

(“北京”， R)

(“京天”，L)

(“天安”，M)

(“安门”，R)

(“门。”，S)

从以上列举出的这些特征中，我们大致可以看到特征的性质不同。如(“北京”，R)，（“天安”，M）就算是比较有信息量的特征。像（“门。”，S）这样的特征，对任务的贡献就比较小，因为“。”一般都作为单字词被标为 S，而不考虑“。”的前面是哪个字。无用特征会导致特征空间膨胀，导致分词器的效率降低。

2.3.2　不同种类特征性能分析

本节的内容中，我们主要分析评价上文中使用的特征模板，包含常用的 unigram 和 bigram 特征。这里我们在 PKU05 上进行实验，分析不同种类特征的分词性能。为了提高实验速度，我们随机抽取了训练集中的 1000 句作为训练集，使用原测试集作为测试集。

U00:%x[-2,0]

U01:%x[-1,0]

U02:%x[0,0]

U03:%x[1,0]

U04:%x[2,0]

U05:%x[-2,0]/%x[-1,0]

U06:%x[-1,0]/%x[0,0]

U07:%x[0,0]/%x[1,0]

U08:%x[1,0]/%x[2,0]

U09:%x[-1,0]/%x[1,0]

U10:%x[0,1]

表 2.1　PKU05 上不同种类特征的分词结果

特征空间	F1 值
U00	0.541
U01	0.666
U02	0.807
U03	0.684
U04	0.542
U05	0.622
U06	0.795
U07	0.807
U08	0.632
U09	0.630
U10	0.504
U02+U06	0.867
全部特征模板	0.878

以上结果可以看出，unigram 特征中最有效的特征是 U02，也就是待分类的位置上的字本身，bigram 特征中最有效的是 U06 和 U07，是该位置向前一个字形成的 bigram 和向后一个字形成的 bigram。另一方面，U02+U06 组成的特征模板所能达到的性能和全部特征模板所能达到的性能相差无几，但特征数量却差好几个数量级。为了便于我们后续开展的分析工作，我们选用 U02+U06 的特征模板，对 unigram 特征和 bigram 特征进行深入的分析。

2.4 使用 $Chi^2_{\max}$ 算法进行特征选择

2.4.1　中文分词任务中的 $Chi^2_{\max}$ 算法

通过第一节的分析可以看到，通过字标注系统，中文分词任务实际上转化成了一个分类任务，句子中的每个位置被分为 LRMS 四类。而文本分类领域有很多成熟的特征选择算法，其中，$Chi^2_{\max}$ 就是最常用的特征选择算法之一。实际上，$Chi^2_{\max}$ 统计量衡量了特征单元 t 和类别 c 的独立程度，这个评价指标在中文分词的任务中也同样适用。如果一个特征和某个类别非常相关，那么往往该特征在分词过程中的贡献比较大，如“我们”的“们”字，本身出现的频度较高，

加之其经常作为一个词的末尾，换句话说，经常和类别“R”相关，那么，“们”字算是一个比较好的特征，另外一个例子，“把”字，在“伞把”里面出现在词的右边界，在“把手”里出现在词的左边界，而且经常作为一个单字词出现，那么单纯从一个“把”字出发，很难推断出词的边界，需要借助于上下文的信息，所以，“把”字这样的特征，在分词过程中的贡献不大。从以上两个例子可以看出，虽然都是 unigram 特征，但是其对于分词任务的重要性是不同的，基于这样的出发点，我们引入了文本分类领域的 $Chi^2_{\max}$ 特征选择算法来对中文分词中的特征进行更加细化地分析。

2.2 节中我们已经介绍了 $Chi^2_{\max}$ 的算法，这里针对中文分词任务，需要对原有的算法进行调整以适应新体系下的评价。首先使用之前选定的特征模板产生特征全集，接着对全集中的每个特征，针对每个类别，我们可以得到四个统计量：A 表示该特征指向特定类别的次数，B 表示该特征指向其他类别的次数，C 表示特定类别中该特征没有出现的次数，D 表示其他类别中不出现该特征的次数，与传统文本分类任务中的定义相同，N 为以上 4 个变量的总和。通过这样的定义，我们可以对每一个特征计算出和任意一个类别之间的开方统计量：

$$Chi^2(t,c)=\frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \quad (2-3)$$

理想情况下，特征 t 和类别标签 c 同时出现或同时不出现，这表明 t 和 c 有很强的关联性。

举例说明以上计算过程，比如下面这个句子：

我/爱/北京/天安门/。

S　S　LR　L M R S

假设使用 U02:%x[0,0]特征模板，即只使用该位置的字本身作为特征，考虑计算特征 Chi^2(“北”，L)的值，“北”字在整句话中出现一次，其出现在一个词的左边界上，所以标签是“L”。整句话中，L 出现了两次，而且“北”并没有指向其他类别，所以，A 的值为 1，B 的值为 0，C 的值为 1，D 的值为 6，根据公式计算所得开方值为 0.97959。

对于特征单元 t，我们可以计算出其与所有类别的开方统计值。我们用其中最大的开方值代表 t 最终的开方值，其定义如下公式所示：

$$Chi^2_{\max}(t)=\max_{1\le i\le m} Chi^2(t,c_i) \quad (2-4)$$

$Chi^2_{max}(t)$的大小即用来衡量该特征单元的重要程度。上例中，我们计算出Chi^2(“北”，L)，Chi^2(“北”，R)，Chi^2(“北”，M)，Chi^2(“北”，S)四个开方值，并取这四个值中最大的作为Chi^2_{max}值，用来衡量该特征的重要性。

2.4.2　特征选择的性能评价

按照上述的计算方法，我们在 PKU05 和 MSR05 上计算了所有特征的开方值。

表 2.2　部分特征的Chi^2_{max}值(PKU05)

本报	3042.427	R
李	2982.954	S
中	2957.176	L
动	2955.435	R
场经	2955.355	M
二十	2944.509	M
%	2913.378	R
界	2907.441	R
世	2890.081	L
共中	2872.73	M
时，	2871.977	S
入	2846.86	R
解放	2840.507	M
他们	2831.8	R

上表列出了 PKU05 部分特征的Chi^2_{max}值，第一列是特征，包含 unigram 特征和 bigram 特征，第二列是该特征的Chi^2_{max}值，第三列是该特征取到最大的Chi^2值的类别标签，比如第一行中，“本报”的类别是“R”，说明“本报”这个特征指向“R”类，即右边界时取得最大的开方值。我们对 PKU05 和 MSR05 上的每一个特征计算其相应的Chi^2_{max}值，并按照该值进行排序，这两个数据集上特征的Chi^2_{max}值分布如下图所示：

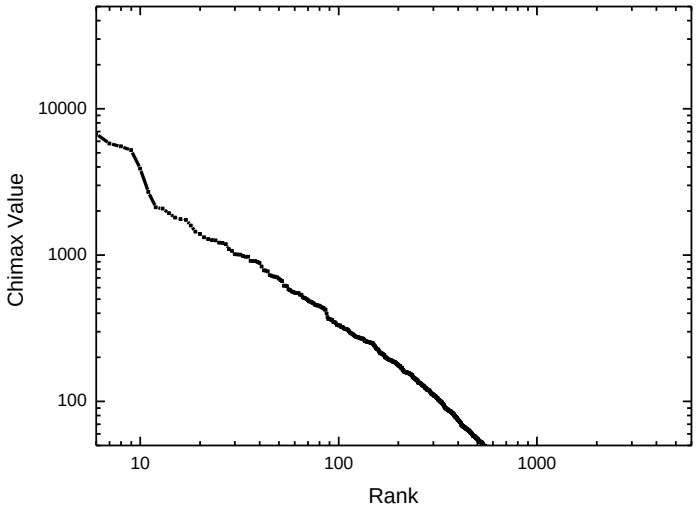

图 2.1　PKU05 上的 Chimax 值分布(对数图)

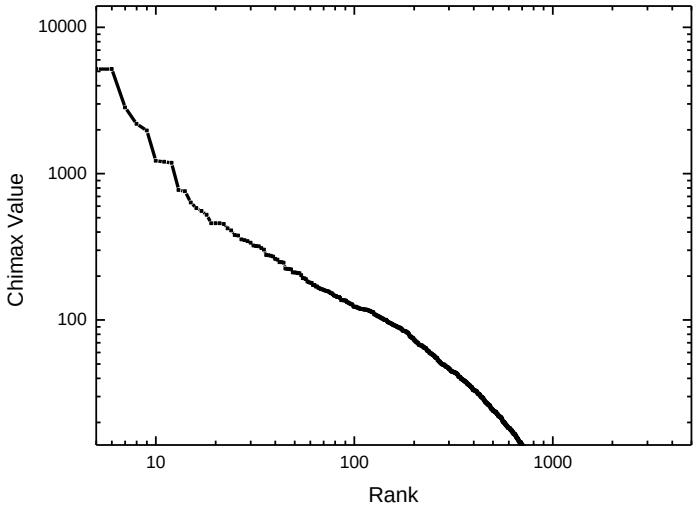

图 2.2　MSR05 上的 Chimax 值分布(对数图)

从以上两幅图中可以看到，两个数据集上的 $Chi^2_{\max}$ 值有着相似的分布。在对数坐标系下，分布曲线呈近似直线，说明其分布大致符合齐夫率。只有很少一部分特征具有较大的 $Chi^2_{\max}$ 值，大部分特征的 $Chi^2_{\max}$ 值都很小。由此推知在分词

过程中，只有一部分特征起着重要的作用，很多特征贡献不大甚至起干扰作用。为了进一步证明我们的结论，进行了以下两组实验，我们研究了 PKU05 上的 unigram 和 bigram 特征，共有 281889 维，使用 $Chi^2_{\max}$ 值对特征进行选择，只用部分特征进行分词实验，考虑到充分性，我们在一组实验中使用 PKU05 训练集训练，PKU05 测试集测试；在另外一组实验中，使用 PKU05 训练集训练，MSR05 测试集测试。我们变换特征维度，观察不同维度下的分词效果，以证明特征选择的有效性，实验结果如下：

表 2. 3　PKU05 训练 PKU05 测试在不同特征维度下的性能

特征维度	Recall	Precision	F1
5000	0.898	0.909	0.903
10000	0.916	0.925	0.920
50000	0.927	0.937	0.932
140000	0.931	0.939	0.935
200000	0.929	0.937	0.933
250000	0.931	0.940	0.935
281889	0.931	0.939	0.935

表 2. 4　PKU05 训练 MSR05 测试在不同特征维度下的性能

特征维度	Recall	Precision	F1
5000	0.850	0.817	0.833
10000	0.862	0.825	0.843
50000	0.869	0.834	0.851
140000	0.870	0.834	0.851
200000	0.870	0.833	0.851
250000	0.871	0.834	0.852
281889	0.871	0.834	0.852

从以上两个表可以看出，特征选择算法是有效的，PKU05 训练集训练，PKU05 测试集测试的情况下，只需要使用 140000 维特征即可以达到最高的分词性能，只使用了原有特征集合的一半，在很大程度上降低了算法的时间复杂度

和空间复杂度。当特征维数取 200000 时，性能反而有所下降，证明了我们的猜想，即排在后面的特征有可能对分词过程起干扰作用。PKU05 训练集训练，MSR05 测试集测试的情况下，同样观察到这样的趋势，在取 250000 维特征时即可达到分词的最高性能。因为 PKU05 和 MSR05 属于非同源数据集，所以需要的特征维数更高，但同样说明特征选择算法在中文分词任务上是适用的。CRFs 是无向图模型的一种形式,它采用了链式无向图结构计算给定观察值条件下输出状态的条件概率，模型本身决定其时间复杂度和空间复杂度非常高，使用特征选择算法，可以在不损失分词性能的前提下，提高分词器的处理速度。

第3章 基于 CRFs 置信度的中文分词后处理

3.1　CRFs置信度的定义

对于 CRFs 给出的每个切分，可以计算得到一个置信度(CRFs confidence)[17]。这个置信度本质上是一个边缘概率，即对于一个特定位置的切分的可能性。这个置信度是介于 0 和 1 之间的实数。切分的置信度可以通过受限的前向-后向算法得到。首先通过标准的前向-后向算法计算得出整个序列的似然度，再通过受限的前向-后向算法计算指定某切分时整个序列的似然度，两者的比值即该切分的置信度。在本文的工作中，我们使用 CRFs 置信度作为后处理工作的切入点。下面几节中我们将对后处理算法进行详细的介绍。

3.2　CRFs置信度的特性

一些汉字倾向于出现在词中的固定位置，而另外一些汉字则没有这个性质。比如说“们”这个字，一般作为词的后缀出现。“的”这个字，大部分情况下作为单字词出现。另一方面，很多汉字到底出现于词中的哪个位置取决于该字所处的上下文。在[3]的工作中，汉字被经验性地分为这两类。

其中，前一类的字在分词任务中提供了非常有用的信息。因为这些字在词中出现的位置相对固定，那么根据该字做出的切分在大部分情况下都是合适的。另外一类汉字可以以相近的概率出现在词中的不同位置，换言之，该分词信息是不可靠的。这样的特征导致的分词结果其 CRFs 置信度比较低。利用第二章中给出的分词特征评价方法，我们可以看到，少量特征起到非常重要的作用，大部分特征对分词过程只起很小或是根本不起作用。这样 $Chi^2_{\max}$ 值排在前面的特征非常重要，在下表中，我们列出了 PKU05 和 MSR05 两个数据集上排在前 20 的特征 (unigram 特征)：

表 3. 1　PKU05 MSR05 上 Chimax 值最大的 20 个特征

	Corpus: SIGHAN05	
Rank	PKU05	MSR05
1	**的**	**的**
2	**和**	**了**
3	**在**	**在**
4	**是**	**和**
5	**了**	**是**
6	与	与
7	业	对
8	为	公
9	等	将
10	者	等
11	将	也
12	对	斯
13	发	就
14	说	说
15	华	尔
16	展	拉
17	泽	业
18	们	他
19	把	们
20	又	个

从上表中，我们可以看到：两个不同的数据集上，其前 20 的特征中间有很多是共同的，这说明这些特征靠前并不是偶然的现象。两个数据集上最重要的 6 个 unigram 特征是一致的，说明这种性质是普遍存在的，不依赖于数据集。这些特征在文本中出现的频度很高，另一方面，它们倾向于出现在词中的固定位置。所以，这些特征为分词过程提供了非常有利的信息，它们可以在很大程度上影响分词器的性能，比如看下面这个例子：

句 1:莫雷/的/手/中/有/一个/苹果/。

句 2:莫雷手/中/有/一个/苹果/。

两个句子的意思一样，但是切分结果却不同。这里我们使用的是 PKU05 训练出的 CRFs 模型进行切分。句二中由于省略了虚词“的”，导致了错误的切分结果。下面我们将更加细致地观察这种现象。

这里是另外一组例子：

句 3:奥巴马麦/凯恩/出席/了/会议/。

句 4:奥巴马/和/麦凯恩/出席/了/会议/。

句 4 被正确地切分，句 3 的切分则是错误的。两个句子的意思一样，但是第二句中省略了虚词“和”，导致了错误的切分结果。下面我们从 CRFs 置信度的角度来分析这一问题。下图是这两个句子的 CRFs 置信度曲线：

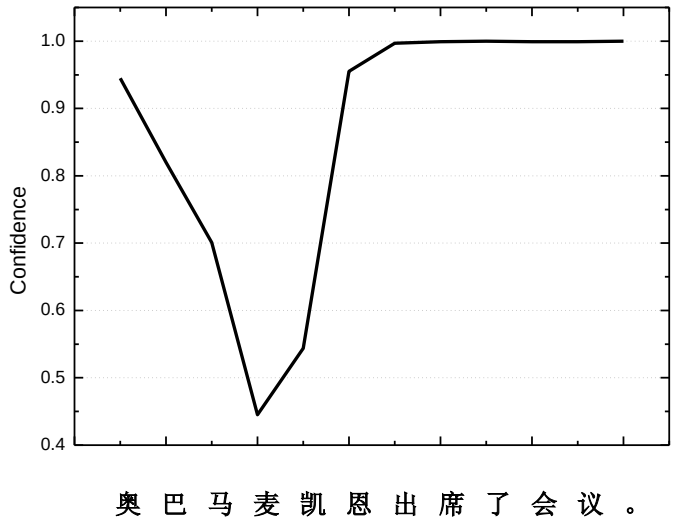

图 3.1　句 3 的置信度曲线

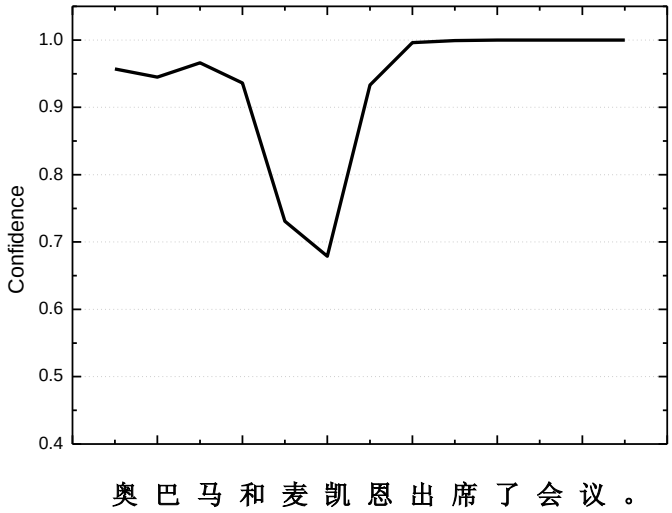

图 3.2　句 4 的置信度曲线

由上面两个图可以看出，“和”字的存在非常重要，该特征提高了整个置信度曲线的值，因为“奥巴马”和“麦凯恩”都是未登录词，如果“和”不存在的话，单纯凭借语料库里的知识难以将其切分正确。这里也可以看出，CRFs 的分词器对于未登录词有比较好的切分结果，也是凭借于一些常见的特征，比如这里的“和”字。一旦这样的特征在句子中被省略，就很容易引入分词错误，单纯依靠 CRFs 分词器是很难解决的。从这个例子，我们看到了 CRFs 分词器的局限性。所以，要想取得更高的分词效果，需要对基于 CRFs 的分词结果进行相应的后处理。

另一方面，类似与“和”这样的高置信度特征有时也会给分词过程带来干扰。比如说下面这个例子：

句 5:他们/家花/了/近/两万/元/。

句 6:他们/家/花/掉/近/两万/元　。

句 6 被正确地切分，句 5 的切分则是错误的。“了”这个特征对它的上下文切分带来了干扰，将两个单字“家”“花”错误地捆绑在一起，组成一个二字词。这种高置信度特征在 CRFs 分词过程中起到非常重要的作用，但因为其有时会给分词过程带来干扰，而 CRFs 自身无法解决这一问题下面我们从 CRFs 置信度的角度来分析这一问题。下图是这两个句子的 CRFs 置信度曲线：

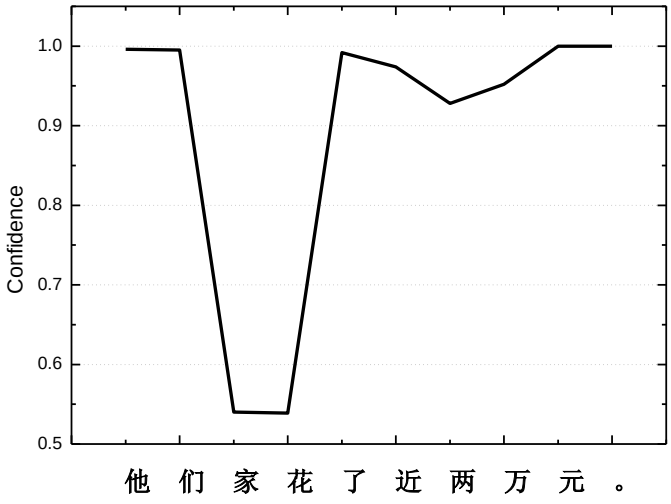

图 3.3　句 5 的置信度曲线

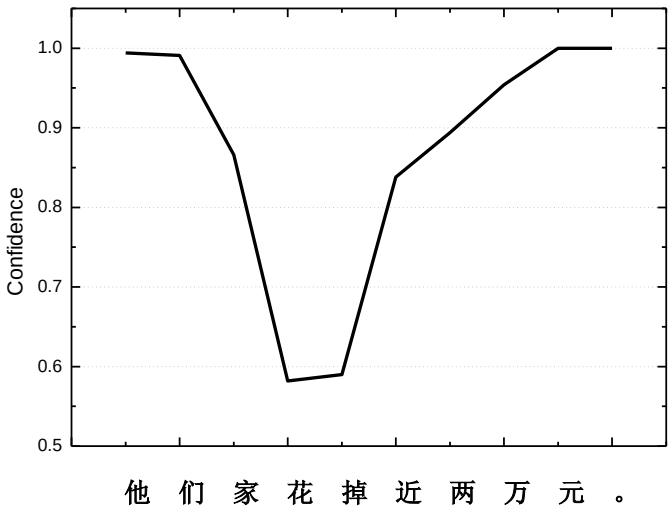

图 3.4　句 6 的置信度曲线

由上面两个图可以看出，尽管“了”是一个高置信度的特征，但是“家花”这个片段的置信度很低，“他们”是高置信度片段，这样“家花”就是处在两个高置信度片段之间的低置信度区间。两个高置信度区间都是分词的边界，受上下文的影响，“家花”被错误地切成了一个词。一方面，CRFs 依赖于高置信度特征，另一方面，高置信度特征也可以引入干扰。基于 CRFs 模型自身的特点，

这样的困难很难依靠 CRFs 自身来解决。这个例子同样反映了 CRFs 分词器的局限性。所以，要想取得更高的分词性能，需要对基于 CRFs 的分词结果进行相应的后处理。

CRFs 置信度为我们的后处理工作提供了非常好的切入点。低置信度区间中出现切分错误的可能性更大，这就为我们的后处理工作指出了目标与方向。低置信度区间的产生，主要有两个原因，一是未登录词，二是出现了比较生僻的搭配，如上面例子中的“家花”序列。关于未登录词的问题，我们在下一章中解决。我们的后处理主要处理生僻搭配一类的问题。从上文的分析中我们看到，CRFs 自身存在固有弱点，由上一章中的分析可见，在 CRFs 分词过程中，少量特征起到了非常重要的作用。这些特征是高置信度特征，利用其信息可以推断其自身以及上下文的切分方案。但是，一方面，在某些语境下，高置信特征会被省略，比如某些用法相对固定的虚词，给分词过程造成了困难。另一方面，有时高置信度特征也会对分词过程造成干扰。CRFs 分词器自身无法克服这样的问题。所以，对 CRFs 分词结果的后处理是必要的。出现未登录词或是出现生僻搭配时，低置信度区间就会出现，在这样的上下文中，出现切分错误的可能性也比较大。我们在 PKU05 和 MSR05 两个数据集上进行了分词错误和 CRFs 置信度之间关系的调研，如下图：

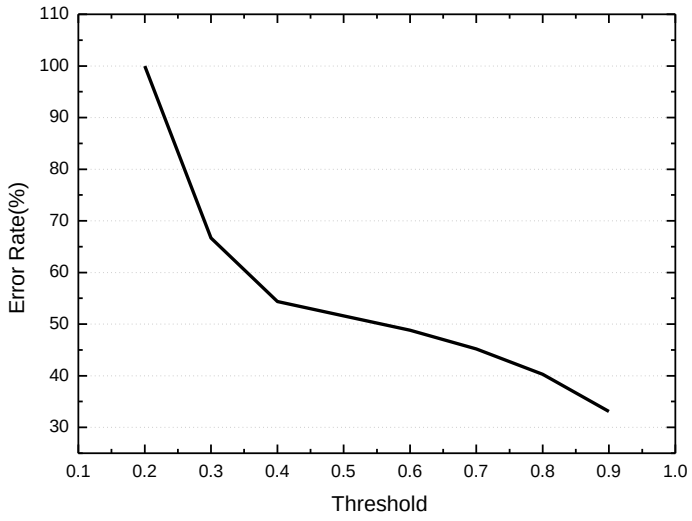

图 3.5　PKU05 上置信度阈值与错误率的关系

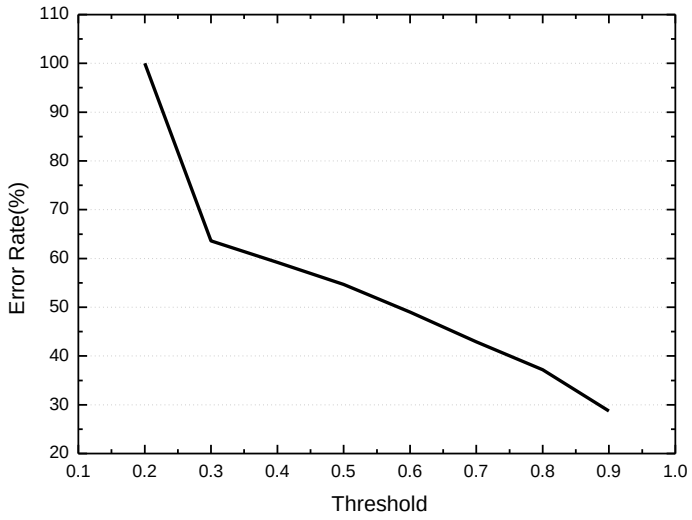

图 3.6　MSR05 上置信度阈值与错误率的关系

以上两图反映了 CRFs 置信度和切分错误之间的关系。这里的阈值是这样定义的，一个分词片段中 (以分词边界为界)，取整个序列中最低的 CRFs 置信度为整个序列的置信度。以上两图中的横坐标为置信度的阈值，以 MSR05 上的曲线为例，当阈值取到 0.9 时，错误率在 30%左右，即 CRFs 置信度小于 90%的所有分词序列中，有 30%的含有分词错误。两幅图中都存在一样的趋势，阈值取得越小，分词错误率越高，这就说明了分词错误率和 CRFs 置信度之间存在反相关关系。这样，我们可以使用 CRFs 置信度作为切入点，对原分词结果进行修正。在下一小节中，我们将具体介绍我们的算法。

3.3　基于CRFs置信度的后处理算法

基于我们以上的分析，CRFs 置信度可以作为我们后处理工作的切入点，因为 CRFs 置信度和切分的错误率呈反相关关系。我们的算法只需对置信度低于设定阈值的片段进行处理，以期对分词错误进行修正。

3.3.1　不同长度的低置信区间

不同长度的低置信区间所包含的切分错误种类不同，当低置信区间的长度较短时，如前分析，如果高置信特征对上下文产生干扰，则有可能将二元组错划为词，如上文中的“家花”。当低置信区间的长度较长时，则很有可能是长词被错误地切开，如成语，机构名等，对于这种情况，我们采用的修正策略就是尽可能地恢复长词。自动分词的一个重要前提是：至少要在计算的意义上清楚界定真实文本中每个词语的边界。在每本汉语语法教科书中，都可以找到有关“词”的一条相当抽象的定义：语言中有意义的能单说或用来造句的最小单位。在计算上，这种模棱两可的定义是不可操作的，或者说，是不可计算的。中文文本是基于字的，对同一段中文文本不同的人可能给出不同的分词方案。前人研究显示，即使在母语为汉语的话者之间，对中文词语的平均认同率只有 0.76 左右[3]。为了研究不同话者对不同长度的词语的认同度，我们进行了如下一组实验：

我们在 YUWEI 语料库中随机抽取了 50 个段落，四个志愿者参加我们的实验，手工对这些段落进行分词。我们用如下的定义来衡量不同标注结果之间的一致性：对四个志愿者标号为#1，#2，#3，#4。首先统计用户标出的词，得到一个词表，假设#1 的用户词表为 A， #2 的用户词表为 B。用户#1 和用户#2 之间的分词一致性就定义为 $\frac{|A\cap B|}{|A\cup B|}$，是介于 0 和 1 之间的实数。

我们对二字词和长于二字的词分别计算四人间的一致性，如下表所示：

表 3.2　二字词的一致性

	#1	#2	#3	#4
#1	1	0.7993	0.8292	0.7893
#2		1	0.8103	0.7922
#3			1	0.8920
#4				1

表 3.3　三字及三字以上词的一致性

	#1	#2	#3	#4
#1	1	0.3374	0.4951	0.3149
#2		1	0.3693	0.3251
#3			1	0.4993
#4				1

实验结果表明关于二字词的一致性要远高于二字以上词的一致性，这也从侧面对我们针对不同长度的低置信区间采用不同的修复措施的可行性。经过 CRFs 切分的语料，其精度已接近或超过 90%，对其进行后处理，稍有不当，就会导致分词性能下降。二字词之间的一致性较强，我们在处理二字长的低置信度单元时，可以采取较为严格的策略，而对二字以上词的修复采用较为保守的策略。

3.3.2　启发式规则

为了提高分词器效率，降低在后处理步骤所耗费的时间，我们采用了词典+启发式规则的方法。这样一可以降低运算时间，二便于扩充资源，只需要增添词表即可。在我们的实验环节里，将词表限制为训练集中的词表，这样可以与 SIGHAN 上现有的封闭测试结果比较。3.2 节中，实验结果说明了分词错误率和 CRFs 置信度之间存在反相关关系。这样，我们可以使用 CRFs 置信度作为切入点，对原分词结果进行修正。上一小节中的实验结果说明二字词之间的一致性较强，我们在处理二字长的低置信度单元时，可以采取较为严格的策略。二字以上词的一致性较弱，经过 CRFs 切分的语料，其精度已接近或超过 90%，对其进行后处理，稍有不当，就会导致分词性能下降。在这种情况下，我们对二字以上词的修复采用较为保守的策略。

使用训练集文本的词表作为资源，对 CRFs 切分的结果进行修正，首先设定阈值，筛选出所有的低置信片段，然后根据其长度的不同，将其划分为三类：第一类，长度为两个字，因为字数较短，受上下文的影响比较强，会出现两种类型的错误，一是将二字词错切为两个单字词，二是将两个单字错误的合并成二元组。这里需要采用严格的修复策略来修复这两种错误。考虑到二字词的一致性较强，训练集中二字词的覆盖度较好，即未登录词大部分是长度大于二字的词，所以我们这里使用训练集词表重新切分，若训练集中有该二字词，则将

两个单字合并为二字词，若训练集词表中没有该二字词，则认为其为错误划分的二元组。这样，前述的“家花”问题可以得到解决。第二类，长度为三个字的低置信片段，“2+1”和“1+2”型歧义是造成三字片段分词错误的主要原因。这里，我们使用词表信息重新切分，若得到的结果与原切分结果不同，则对结果进行修正，否则，保留原分词结果不变，这种策略要比对二字词的处理保守，当词表中不包含该词时，维持原有的切分方案不变。第三类，四字及四字以上词的低置信片段，我们采用最为保守的修复措施，只有词表信息支持将整个片段切分为一个词时，才对之前的分词结果进行变更。我们将在下一节中通过实验证明算法的有效性。

3.4　实验

3.4.1　实验设计

实验环节中我们使用了三个数据集：PKU05，MSR05 和 YUWEI 语料库。三个语料库的统计数字如下：

表 3.4　PKU05 MSR05 和 YUWEI 的统计数字

Corpus	Encode	Train set (words)	Test set (words)
PKU05	GBK	1.1M	104K
MSR05	GBK	1.37M	107K
YUWEI	GBK	2.4M	59K

在我们的实验环节里，将词表限制为训练集中的词表，这样可以与 SIGHAN 上现有的封闭测试结果比较。实验中使用了 Taku Kudo 提供的 CRFs 工具包，版本是 0.51。我们使用了前文介绍的 LMR 系统。特征模板使用了 unigram 和 bigram 五字窗口特征，其它参数取默认值。在取低置信度区间时，需要设定 CRFs 置信度的阈值。

3.4.2　实验结果

在 YUWEI 语料库上，我们改变该阈值以取得不同的分词效果，结果如下：

表 3. 5　YUWEI 在取不同置信度阈值时的结果

Threshold	Recall (%)	Precision (%)	F1 (%)
90%	94.664	94.824	94.734
80%	94.876	95.098	94.987
70%	**94.901**	**95.149**	**95.025**
60%	94.848	95.117	94.982
50%	94.670	94.987	94.828
40%	94.563	94.974	94.768
baseline	94.506	94.982	94.743

从上表的结果我们可以看出，与 baseline 相比，进行后处理可以提高分词性能。当阈值在 40%到 90%之间变动时，都能取得比原有的基于 CRFs 的分词算法高的性能，这说明我们的后处理方法是有效且稳定的。YUWEI 的结果显示阈值取 70%时可能达到最高的分词性能。在后续的实验中我们将沿用该经验值。PKU05 和 MSR05 上的实验结果如下所示：

表 3. 6　后处理算法与原算法性能上的比较

Corpus		Recall	Precision	F1
YUWEI	baseline	0.945	0.949	0.947
	our	**0.949**	**0.951**	**0.950**
PKU	baseline	0.942	0.953	0.947
	our	**0.947**	**0.955**	**0.951**
MSR	baseline	0.959	0.965	0.962
	our	**0.963**	**0.967**	**0.965**

实验结果显示后处理算法可以有效提高分词的正确率，该算法在不同来源的数据集上均有效。分词的召回率和准确率同时得到提升。

表 3. 7　PKU05 MSR05 在封闭测试上的最好结果

SIGHAN05 best results (close)			
Corpus	Recall	Precision	F1
PKU05	0.946	0.954	0.950
MSR05	0.962	0.966	0.964

本文的工作中我们只使用了来源于训练集中的词表来进行后处理，所以我们将 PKU05 和 MSR05 上的结果与这两个数据集上封闭测试的最好结果 (见上表) 进行比较，实验结果显示后处理算法可以提供更好的分词性能，该算法是有效的。

3.5　结论

在本章中，我们分析了 CRFs 置信度的特性和 CRFs 分词器的局限性，CRFs 的分词器对于未登录词有比较好的切分结果，也是凭借于一些常见的特征。一旦这样的特征在句子中被省略，就很容易引入分词错误，单纯依靠 CRFs 分词器是很难解决的。另一方面，高置信度特征也可以引入干扰。PKU05 和 MSR05 上的分析工作说明了分词错误率和 CRFs 置信度之间存在反相关关系。这样，我们可以使用 CRFs 置信度作为切入点，对原分词结果进行修正，以克服 CRFs 分词器自身的弱点。

不同长度的低置信区间所包含的切分错误种类不同，当低置信区间的长度较短时，如前分析，如果高置信特征对上下文产生干扰，则有可能将二元组错划为词。当低置信区间的长度较长时，则很有可能是长词被错误地切开，如成语，机构名等，对于这种情况，我们采用的修正策略就是尽可能地恢复长词。

实验结果显示后处理算法可以有效提高分词的正确率，该算法在不同来源的数据集上均有效。分词的召回率和准确率同时得到提升。

第4章 利用篇章信息识别未登录词

4.1　背景介绍

从中文分词任务的发展现状来看，目前，中文自动分词的歧义切分技术已达到了较高的水平，未登录词问题仍不容乐观。未登录词引发的分词错误数量比歧义引发的分词错误数量要大 5 倍以上[27]。基于 CRFs 的分词器可以达到 state-of-art 的分词效果，所以本文中的研究工作主要建立于 CRFs 分词器基础上，但是，面向开放语料的分词性能仍然不能令人满意。随着 Web 上的中文语料爆炸性的增长和对中文语料自动分词的需求不断加强，面向 Web 的中文自动分词技术逐渐引起了人们的重视。和传统的中文自动分词不同的是，Web 环境下的分词面临着更为复杂的问题。由于 Web 上的文本变化较为剧烈，每时每刻都有新的文本产生，旧的文本消亡。预先训练好的分类模型可能并不适应于 Web 的环境。训练语料只能覆盖有限的一部分词语，加之新的词汇不断地在 Web 文本中涌现，OOV 在面向 Web 的中文分词处理中成为非常严峻的一个问题。

未登录词语粗略地可分为两类，一是各种规则的专名类别，包括规范的人名，地名，机构名等；二是不规范的名称，如绰号，外来人名，各种新的专业术语，缩略语等。第二种不规范的未登录词在 Web 文本中尤为普遍，给目前的分词系统带来了极大的困难。

在这一章中，我们将详细介绍这些方面的内容，并提出针对未登录词问题的解决方案。

4.2　文本段落中的信息

基于 CRFs 的分词器在遇到未登录词时，往往会产生切分不一致的问题，对分词的精度影响较大。比如下面这个例子：

冬虫/夏/草到底*/是/虫/还/是/草/冬天/是/虫/，/夏天/是/草/，/****冬虫/夏/草****/是/个/宝/。/****冬虫/夏草****/简称/虫草/，/是/冬季/真菌/寄生/于/虫草/蛾幼/虫体/内/，/到/了/*

*夏季/发育/而/成/。/**冬虫/夏/草**/因/此/得/名/。*

我们使用 PKU05 训练集训练出的 CRFs 模型得到上述的分词效果，“冬虫夏草”并没有在训练集中出现，因此是未登录词，上段文字中主要的分词错误就集中在“冬虫夏草”上，短短一段文字中，同样的串，却有截然不同的三种切分方案：

冬虫/夏/草到底/

/冬虫/夏/草/

/冬虫/夏草/

由此可见，CRFs 的分词器在遇到未登录词时，往往会产生切分不一致的问题，对分词的精度影响较大。

另一方面，上段文字又给我们这样的启示：部分未登录词倾向于在一段上下文中反复出现，我们可以利用篇章中的信息，用基于串频的方法，识别出这类词，从而对原有的分词结果进行修正。但是，单纯基于串频，很容易引入各种噪声。经过 CRFs 切分的语料，其精度已接近或超过 90%，对其进行后处理，稍有不当，就会导致分词性能下降。所以，单纯基于串频的方法，对 CRFs 分词结果的修正是不适用的。如何解决这样的问题，我们将在下一章中介绍。

4.3　利用CRFs置信度识别未登录词

未登录词对分词器的性能有很大的影响。上一小节中，我们注意到了这样的现象，部分未登录词倾向于在一段上下文中反复出现，换句话说，其在局部的频度较高，我们可以通过设定频度的阈值来筛选出这样的串，然后根据该信息对分词结果进行修正。但是，并不是所有高频字串都是未登录词，这样做会产生较大的干扰。如何更高精度地识别符合要求的高频串？我们这里使用 CRFs 置信度来解决这一问题，首先回顾第三章中提到的一个例子：

句 1:奥巴马麦凯恩出席了会议。

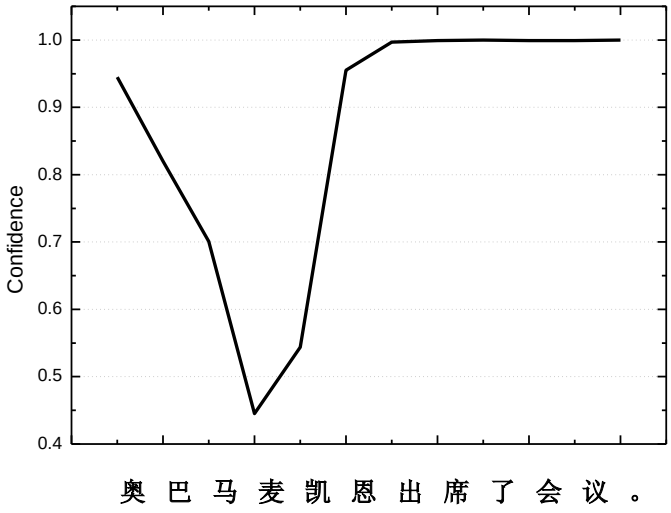

图 4.1　句 1 的置信度曲线

当未登录词出现时，CRFs 置信度会降低。上图也可以验证这一点，“奥巴马”，“麦凯恩”并没有出现在 PKU05 的训练语料中，故都属于未登录词，所以关于该片段的切分置信度较低。究其原因，CRFs 置信度是和该特征在训练集中出现的频度相关的，未登录词并没有在训练集中出现，所以关于它们的切分置信度一般较低。低置信度区间中出现切分错误的可能性更大，未登录词问题会导致分词器性能的下降。虽然 CRFs 可以帮助我们识别未登录词，低置信度区间的产生，主要有两个原因，一是未登录词，二是出现了比较生僻的搭配。也就是说，单纯依靠 CRFs 来筛选未登录词，同样会引进噪声。因为 CRFs 的后处理对噪声非常敏感，所以单纯使用 CRFs 来筛选同样是不可行的。为了提高识别的精确率，我们采用了这样的算法，先使用 CRFs 置信度来确定候选片段，再基于上下文的信息进行二次筛选，关于二次筛选的方法我们将在下一小节中介绍。这里我们仍然沿用第三章中的方法，选择出低置信区间，留待后续处理。

4.4　重复字串抽取

在上一小节中，我们利用 CRFs 置信度来筛选出候选片段，另一方面，上文的分析指出，部分未登录词倾向于在一段上下文中反复出现，我们可以利用篇章中的信息，用基于串频的方法，识别出这类词，从而对原有的分词结果进行

修正。这里，我们把两种方法结合起来进行筛选，以期达到更高的识别精确率。

在抽取重复字串的过程中，我们使用了后缀树 (PAT-tree) 数据结构。这里，我们简单介绍一下关于后缀树的背景知识。后缀树本质上是一种压缩的二叉查询树，是在信息检索领域被广泛使用的一种高效的数据结构。它将关键字作为二进制位串记录在树的结构中，从根节点到叶节点的每一条路径都代表一个关键字位串。在我们的任务中，叶节点表示的是文本中的字串。在一个长度为 n 的文本中，从任何位置开始到文本结束都是一个字串，即后缀字符串。所有这样的串都在后缀树中进行存储。利用后缀树数据结构可以很容易地提取出频繁字串。

这里我们将使用 CRFs 置信度筛选出的低置信片段转化为后缀树，并提取出频繁字串。这里还面临如下问题，产生的某些字串只是新词的子串，而并不是新词本身，如果界定新词的范围，是下一步要考虑的问题。

这里，我们使用上下文熵这一概念来解决该问题。从直观的角度来讲，有意义的新词语，不但在篇章中重复出现，而且倾向于在不同的左右上下文中出现。上下文熵体现了该字串独立成词的能力和使用上的自由度。

设 $\alpha=\{a_1,a_2,...,a_n\},\beta=\{b_1,b_2,...,b_n\}$ 分别为候选字串 ω 在篇章中的左右上下文集合，字串 ω 在语料库中的上下文熵分布定义为

$$CE_l(\omega)=-\frac{1}{n}\sum_{a_i\in\alpha}C(a_i,\omega)\log\frac{C(a_i,\omega)}{n} \qquad (4-1)$$

$$CE_r(\omega)=-\frac{1}{n}\sum_{b_i\in\beta}C(\omega,b_i)\log\frac{C(\omega,b_i)}{n} \qquad (4-2)$$

其中，$n=\sum_{a_i\in\alpha}C(a_i,\omega)=\sum_{b_i\in\beta}C(\omega,b_i)$，$C(a_i,\omega)$ 和 $C(\omega,b_i)$ 分别为字串 $a_i\omega$ 和 ω,b_i 在篇章中的出现次数。从以上的定义可以看出，上下文熵越小，说明该字串更倾向于出现在固定的上下文中，其更有可能是新词的一部分。下面我们举例说明具体的识别算法，如上文中所举的例子中，考察“冬虫夏”这个字串，在它的右端每次都是紧跟“虫”字，这样它的下文熵为 0，故说明其为词的一部分，需要对其扩充，扩充为“冬虫夏草”这个串以后，其下文出现了四个不一样的字，这样它的下文熵变为 2，故不再对其扩充，这样也就正确识别出了该词的右边界，对左边界可以采用相同的算法来推断。

综上，我们对于未登录词识别的算法按照如下流程：首先使用 CRFs 分词器

预切分，得到每个切分的 CRFs 置信度。按照设定的阈值，选择出低置信度区间。第二步，将选择出的片段转化成后缀树，统计高频字串。第三步，对每个高频字串计算其上下文熵，若上下文熵小于设定的阈值，则对该字串进行扩充，该算法反复迭代，直到字串不再变化为止。最后得到的字串即为识别出的未登录词，利用这个信息对原分词结果进行修正，得到新的分词结果。在下一节中，我们将用实验证明我们的算法可以提高分词结果的正确率。

4.5　实验

在实验环节我们遇到了缺少评测数据集的困难，目前还没有公开的面向 Web 的篇章分词语料，为了检测上文中的算法，我们构建了一个小规模的测试语料，来源于 Web 文本，涉及体育，机械，旅游，医药等多个主题，含有 4000 多个词。这些语料全部采用手工标注分词结果，作为评测标准。Baseline 采用基于 CRFs 的分词器，比较的结果如下表所示：

表 4.1　带 OOV 识别的分词算法与原算法性能上的比较

	Recall	Precision	F1	OOV recall
Baseline	0.952	0.917	0.934	0.774
带 OOV 识别的分词器	**0.979**	**0.974**	**0.977**	**0.909**

从上表的评测结果可以看出，我们的算法是有效的，由于加入了未登录词识别环节，有效地识别出了文本中的未登录词，有效提高了未登录词的召回率，从而提高了整体的分词召回率，另一方面，由于识别出了未登录词，未登录词的边界可以清楚界定，也进一步提高了分词的准确度。这样，整体的分词性能都有所提高。我们在构建数据集的过程中，使用了不同来源，不同主题的文本，实验结果证明该算法不依赖于文本主题，可以有效地进行篇章分词。

在下文中，我们将给出一些具体的例子，来进一步观察实验结果，以上算法已经实现，目前在清华大学分词器 1.1 的版本中使用，以下为具体的分词结果：

图 4.2　旅游语料分词实例

上图实验中，待切分的语料是有关景区的介绍，文本中间含有很多具体的景点名，都属于未登录词，给分词任务造成了很大困难。基于 CRFs 置信度和篇章信息的识别算法可以有效区分这些词，避免产生干扰，上图中“新词发现”栏给出了识别结果。

图 4.3　航空技术语料分词实例

上图实验中，待切分的语料是关于航空技术的语料，文本中间含有很多术语，有些术语非常生僻，如“高函道比”，分词难度很大。上图中可以看出，我们的算法有效的识别出了这些术语，排除了对分词结果的干扰，取得了较好的分词效果。

4.6　结论

在本章中，我们介绍了基于 CRFs 置信度和篇章上下文信息的未登录词识别算法，并使用该算法对分词结果进行后处理，以期提高分词性能。我们对于未

登录词识别的算法按照如下流程：首先使用 CRFs 分词器预切分，得到每个切分的 CRFs 置信度。按照设定的阈值，选择出低置信度区间。第二步，将选择出的片段转化成后缀树，统计高频字串。第三步，对每个高频字串计算其上下文熵，若上下文熵小于设定的阈值，则对该字串进行扩充，该算法反复迭代，直到字串不再变化为止。最后得到的字串即为识别出的未登录词，利用这个信息对原分词结果进行修正，得到新的分词结果。实验结果证明我们的算法可以有效地识别出上下文中的未登录词，提高分词结果的正确率。

第5章 基于 CRFs 的短文本分类

5.1　背景介绍

近些年来，利用计算机的强大功能对文本进行自动快速的分类引起了研究人员的重视[6, 7]。给定预先定义好的分类体系，以及人工标注好的训练集文档，通过选择合适的机器学习方法，我们可以得到训练好的模型，将这个模型应用于未标注的测试集样本，完成整个自动分类的过程。 短文本分类问题作为文本分类问题的一个分支，除具有共性之外，还面临一些特殊问题需要解决，因为文本长度短，特征稀疏，难以衡量短文本之间的相似性，单纯地从普通文本分类任务中移植的算法有时并不能得到很好的效果[31]。但是，短文本分类任务在 web 环境下得到了越来越广泛的应用。如何有效地提高短文本分类准确度，已成为一个至关重要的问题。

在某些文本处理任务中，由于计算资源的限制，无法对全文进行处理，转而对文章中比较重要的部分，如标题、关键字等进行处理，这里就要用到短文本分类技术。另一方面，随着搜索引擎的广泛应用，用户查询日志分析的重要性日益显现。同时，搜索引擎自身需要对用户的查询进行分类和鉴别[32]。这些应用场合都需要短文本分类技术。

由于短文本分类任务的特殊性，传统的文本分类方法在该领域表现不佳。因为文本较短，面临特征稀疏的问题，文本之间很少含有相同的特征，这样，文本之间的相似性不好度量，这些特殊的性质为短文本分类任务增添了极大的困难。

前人的工作中，很多方法用来克服短文本分类中的特征稀疏问题。这些方法使用外部资源来增加文本之间共享的特征。搜索引擎返回的摘要等信息被用来丰富文本的上下文[33, 34, 35]，在线的数据库，例如 wikipedia 和 ODP (Open Directory Project)也被作为外部资源引入短文本分类任务中[36, 37, 38]。外部资源的引入可以提高短文本分类的准确性。但是，外部资源的使用，比如说利用搜索引擎返回的结果，是非常耗时的步骤，所以，这些方法在对时间要求比较高的场合下，比如说实时系统中，是不合适的。

为了提高分类系统的速度，本章的工作中并没有使用外部资源，而是利用短文本自身的特性来提高分类准确率。因为短文本的长度有限，所以主题集中，短文本的特征之间具有较强的一致性。本章的工作主要包含两方面的主要内容：

其一，对短文本中的特征一致性现象进行研究。其二，在研究的基础上，提出了基于 CRFs 的短文本分类算法，实验证明，基于 CRFs 的算法可以达到更好的分类准确性。

5.2　短文本特征的一致性

在本章的内容中，我们研究了短文本自身的性质。从直觉上来讲，短文本由于篇幅限制，涉及到的话题比较集中。所以，短文本中的特征也比较倾向与涉及同样的主题，这也被称为短文本特征的一致性。随着文章长度的增大，这种一致性会随之减弱。在本节中我们将对以上性质进行定量的分析。

5.2.1　关于短文本特性的定性分析

由于长度限制，短文本通常只有少量的特征，所以，短文本之间很少能有共同的特征，这也造成了短文本分类任务中的特征稀疏问题。但是，从另一方面来讲，短文本也有其自身特殊的有利性质，可以用于短文本分类任务中。短文本通常集中于一个主题，这点与长文本不同。在一篇长文本中，可能有多个片段分别覆盖不同的主题。比如说，一篇关于体育的长文本中，很有可能一段在讨论足球，而另一段在讨论历史。在短文本中则很少会遇到这样的问题。

短文本通常集中于一个主题，这样，短文本中的特征通常也具有很强的相关性。如果一个特征和主题 A 有关，那么它旁边的特征也有很大可能和主题 A 相关。在本文中，我们用“一致性”来指代短文本特征之间的这种性质。

在[39]中，作者提出了一种基于片段的文本分类方法，即先将待分类的文本划分为不同的文章片段，然后对每个片段进行分类，最终利用每个片段的类别信息对整个文章的类别进行推断。这种方法可以提高文本分类的准确度。这种基于片段分类的方法也启示了我们，文章片段，即短文本具有特殊的性质，如果利用得当，可以提高分类的性能。

在下一小节中，我们将定量分析不同长度文本的一致性问题。从直觉上讲，这种一致性与文章长度呈一种轻微的负相关关系。更明确的说就是，文章越长，这种一致性就越不显著。相邻的特征倾向于只与相同的主题有关，随着长度的增加，涉及的主题就会逐渐改变。

5.2.2　关于短文本特性的定量分析

在本小节中，我们在中文大百科语料上进行实验来证明我们关于短文本一致性的假设。中文大百科语料包含 71674 篇文本，分为 55 类，每篇文章都是单

标记[40, 41]。我们的实验主要关注不同长度文本下特征的一致性情况，实验的具体方法如下：

首先，我们在整个大百科语料库上随机选取 500000 个短文本片段，每个片段包含 20 个汉字，接下来，我们把每个片段按照如图所示的方法分为两个部分：NEAR 部分和 FAR 部分，其中，每个部分的长度都是 10 个字：

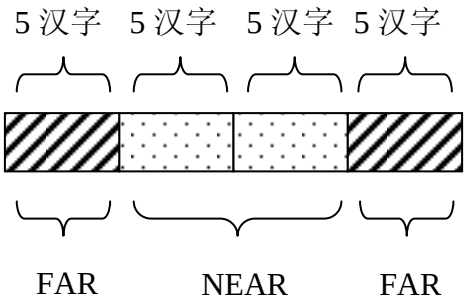

图 5.1　NEAR 部分和 FAR 部分的定义

通过这样的算法，我们得到了 500000 个 NEAR 部分和 500000 个 FAR 部分。为了便于下面的计算，我们定义 A 为所有 NEAR 部分中所包含的不同特征数，定义 B 为所有 FAR 部分中包含的不同特征数。

由上述的计算过程可知，NEAR 和 FAR 部分的长度相同，所以 A 和 B 应该也相差不大。我们独立进行了四次实验，每次都随机选取 500000 个片段，每个片段包含 20 个汉字。实验结果如下表所示：

表 5. 1　大百科上的实验结果

Round	A	B	B-A	(B-A)/A
1	2565451	3144164	578713	0.226
2	1760231	2151481	391250	0.222
3	1885561	2328544	442983	0.235
4	2083919	2565040	481121	0.231

在上表中，我们使用(B-A)/A 来衡量 A 与 B 之间的差别。如果短文本特征没有一致性，那么既然 NEAR 部分和 FAR 部分包含的字数相同，A 和 B 的值应该相差不大。换句话说，(B-A)/A 应该是一个很小的值。但实验结果却显示 A 和 B 之间有很显著的差别。FAR 部分包含更多的特征，因为 FAR 部分被分为两段，这两段是不邻接的，所以这两段倾向于包含不同的特征。这也是 FAR 部分

包含更多特征的原因。NEAR 部分可以被视为是短文本，大部分短文本都集中于一个主题，短文本中的特征也倾向于关联同一个主题。所以 NEAR 部分包含更少种类的特征。实验证明，随着文本长度的改变，特征之间的一致性在改变，两者确实存在反相关关系。短文本中的特征是具有一致性的。

5.3　基于CRFs的分类器

基于我们上一节的分析，大多数短文本集中于一个主题，短文本中的特征具有一致性。如果一个特征与某个主题发生关联，那么其上下文中的特征也倾向于该主题。实验证明短文本的一致性比较显著，该特性如果能被利用，我们可以进一步提高短文本分类的准确度。CRFs 模型被用来在特征之间增添约束。

在本节中，我们将介绍基于 CRFs 的分类器。首先，让我们简要地回顾一下 CRFs 模型，接下来，我们利用了中文分词领域中广泛使用的字标注算法来讲短文本分类问题转化为一个序列标注问题。经过这样的转化，CRFs 就可以用于我们的任务。最后，我们将介绍基于 CRFs 的短文本分类方法。

5.3.1　链式CRFs

在这一小节中，我们简要回顾一下链式 CRFs 算法。CRFs 在中文分词领域得到了广泛的应用。CRFs 是无向图模型的一种形式,它采用了链式无向图结构计算给定观察值条件下输出状态的条件概率[17]。

在短文本分类任务中，文本的特征是可以被观察到的变量，而特征相关的标签是隐藏的变量。从链式 CRFs 的图结构中我们可以看出，隐藏节点之间存在约束，也就是特征相关的标签之间存在约束。我们在上一节中的分析指出，相邻的特征倾向于关联同一个主题。从这个角度上来看，CRFs 是适用于短文本分类任务的。这也是我们是使用该算法的原因。但是 CRFs 是一种序列标注算法，所以无法直接应用在我们的短文本分类任务中。所以，我们需要将短文本分类任务转化为序列标注任务。在这里，我们借用了中文分词领域广泛使用的字标注方法来解决这个问题。在下一小节的内容里，我们将简要介绍字标注方法。

5.3.2　字标注方法

在该小节中，我们借用中文分词领域的字标注方法来解决我们的问题。字标注方法可以用来将中文分词问题转化为序列标注问题。我们使用这个方法将短文本分类问题也转化为序列标注问题，这样 CRFs 算法可以在我们的任务中得到应用。

LMRS 是一种常见的标注方法[3]，这里我们简单回顾一下。每个汉字被打上一个标签，标签是 LRMS 之一。L 代表词的左边界，R 代表词的右边界，M 代表词的中间，S 代表单字词。经过这样的标注，中文分词问题就可以转化为一个序列标注问题，我们给出一个简单的例子如下:

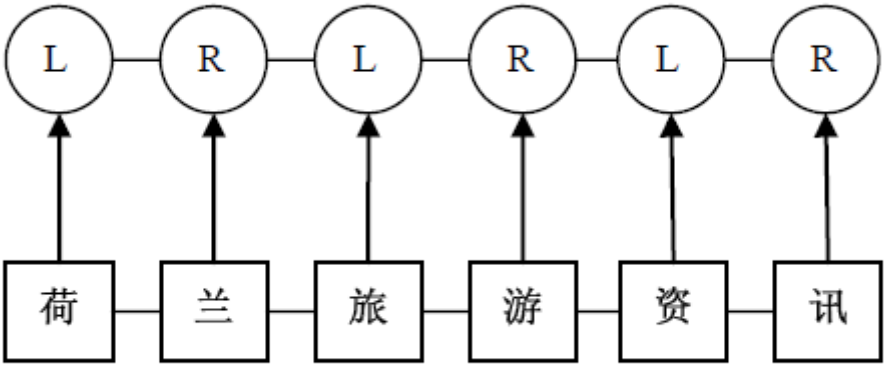

图 5.2　字标注方法

上图给出了关于字标注方法的一个例子，图中方形的节点表示汉字，圆形的节点表示每个汉字相应的标签，“荷兰旅游资讯”这个文本片段被分为三个词: “荷兰”，“旅游”，“资讯”。可以看到，标签之间同样存在约束。比如，M 不能直接跟在 R 的后面。 我们同样使用这种方法将短文本分类问题转化为序列标注问题。

5.3.3　基于CRFs的短文本分类算法

在该小节中，我们使用上文提到的字标注方法。该方法同时用于训练阶段和测试阶段，在中文分词任务中，每个汉字依照其在词中出现的位置给予标签。我们的算法依照短文本的类别对每个字打标签。

下面我们距离来说明基于 CRFs 的短文本分类算法，假设我们的训练集里有这样一篇短文本“荷兰旅游资讯”，该短文本属于“旅游” (Travel) 类，在图中我们用“T”来指代。依照我们的算法，该短文本中的每个字的位置都被标为(“T”), 这样原来的短文本就被转化为一个标注后的序列，如下所示：

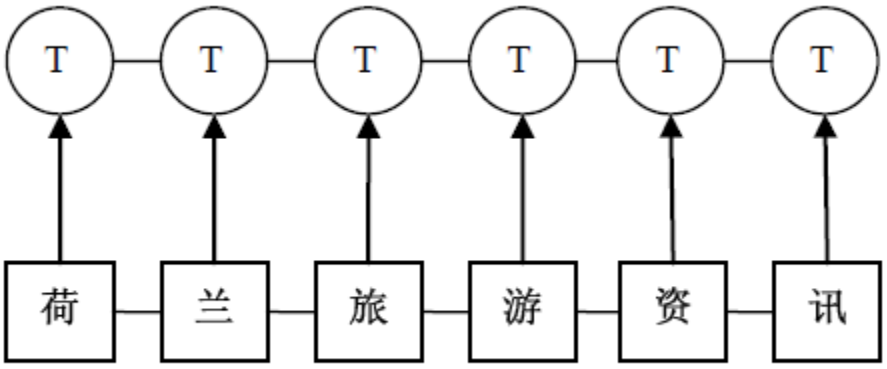

图 5.3　训练步骤中的带标注序列

在上图中，方形节点表示汉字，圆形节点表示该汉字对应的标签。我们使用字标注方法将训练集中原始的短文本转化成了已标注的序列，这样，CRFs 算法可以用来训练模型。得到的模型可以在测试阶段使用。假设在测试集里有一篇这样的短文本“欧洲杯赛程”，我们将使用 CRFs 算法来推断该文本的类别。首先我们将该文本转换为一个待标注的序列，如下图所示：

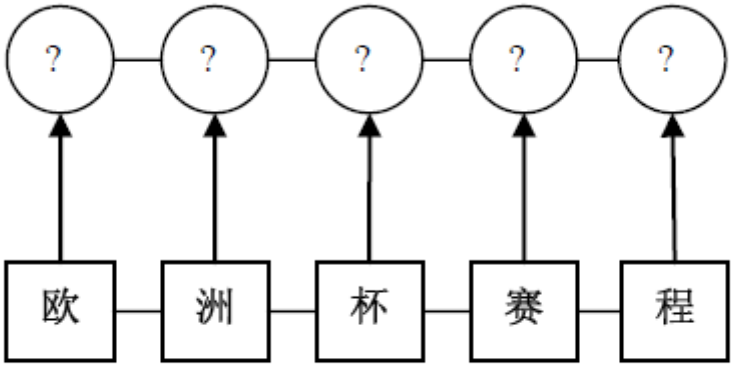

图 5.4　将测试文本转化为无标注序列

第二步，我们使用从训练集得到的 CRFs 模型来对该序列进行标注，其结果如下图所示 (S 代表体育类) ：

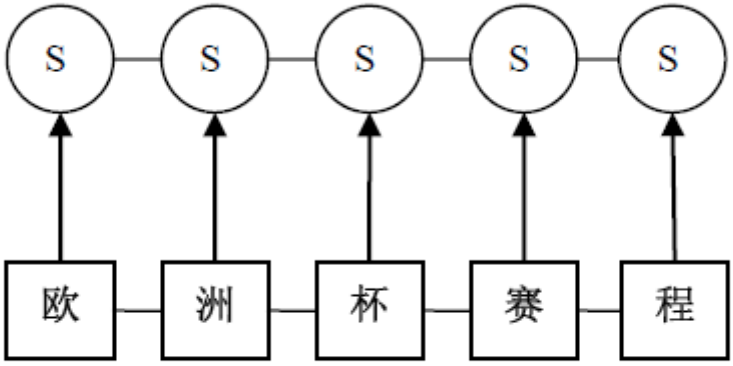

图 5.5　测试步骤中的带标注序列

在上图中，我们通过序列的标签得到了该短文本的类别，该短文本属于“体育”类。在 5.2 节中，我们的分析说明特征之间的一致性随着文本的长度改变，一致性与文本长度之间是反相关关系。当文本长度非常短时，特征之间有较强的一致性。从链式 CRFs 的结构图中我们可以看到，隐藏变量之间存在约束，该模型与短文本特征相符合，所以基于 CRFs 的短文本分类方法可以提高分类的准确度，在下一节中，我们将用实验结果来证实这个结论。

5.4　四个数据集上的实验

我们在四个数据集上检测基于 CRFs 的短文本分类方法，其中，Sohu 和 Netease 短文本数据集是中文语料集，由 web 上的文章标题组成。Ohsumed-all 是英文语料库，由医学摘要构成。我们还使用了 Sogou 搜索引擎的查询日志集合来检测我们的算法在违禁条目检测中的性能表现。在我们的实验中，使用 Unigram 和 Bigram 特征。关于短文本，目前并没有明确的定义，所以在我们的实验中，我们使用经验值“10”来限制文本的长度。在中文文本中，限制为 10 个汉字，英文语料中，限制为 10 个单词。我们将基于 CRFs 的分类算法和传统的基于 SVM 的分类器进行比较。

5.4.1　中文短文本分类实验

本小节的实验使用 Sohu 和 Netease 短文本数据集，其中，Sohu 语料库是公开数据集，共分为 15 类，包含 429819 篇单标记的文本。我们使用 web 文档的标题库作为短文本数据集，经过预处理，数据过滤 (滤除过长的标题) 后，我们

最终得到分别属于 15 类的短文本 65484 篇。我们的实验包含 6 分类任务，10 分类任务和 15 分类任务，在 6 分类任务中，使用的类别有“learning”， “mil”，“news”，“sports”， “travel” 和 “women”。在 10 分类任务中，还使用了“career”，“auto”，“health” 和 “IT”类。在 15 分类任务中，所有短文本都被使用。我们使用不同的比例在随机切分训练集和测试集。基于 CRFs 的分类器和基于 SVM 的分类器性能比较如下表所示：

表 5. 2　Sohu 语料库上的实验结果(训练集占 70%)

	SVM	CRFs
6 classes	0.817	**0.884**
10 classes	0.787	**0.819**
15 classes	0.882	**0.894**

表 5. 3　Sohu 语料库上的实验结果(训练集占 50%)

	SVM	CRFs
6 classes	0.791	**0.851**
10 classes	0.751	**0.785**
15 classes	0.870	**0.883**

表 5. 4　Sohu 语料库上的实验结果(训练集占 30%)

	SVM	CRFs
6 classes	0.760	**0.827**
10 classes	0.700	**0.754**
15 classes	0.852	**0.870**

从上面的实验结果可以看出，我们的方法可以提高分类的准确率。为了进一步的比较两种算法，我们进行了一系列实验，使用不同的训练集与测试集的划分比例，在上述 6 分类问题上的性能比较如下所示：

表 5. 5　Sohu 语料库上不同训练测试比例下的性能比较(6 分类问题)

Train set proportion	SVM	CRFs
0.9	0.824	**0.887**
0.8	0.817	**0.884**
0.7	0.809	**0.871**
0.6	0.800	**0.859**
0.5	0.791	**0.851**
0.4	0.781	**0.842**
0.3	0.760	**0.827**
0.2	0.724	**0.796**
0.1	0.679	**0.750**

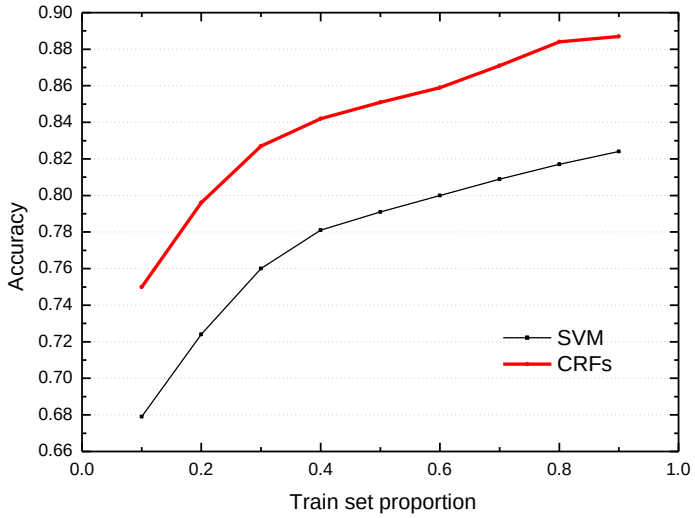

图 5.6　Sohu 语料库上不同训练测试比例下的性能比较

如上所示，实验结果证明我们的方法是有效的，可以显著提高中文短文本分类的准确率，不同条件下的实验结果均说明了这一点。当训练集规模非常小或是文本长度非常短时，我们的方法仍能够提供较好的分类性能。在 Sohu 语料库上的所有实验结果表明，我们的方法可以提供比 SVM 分类器更高的分类准确度。

Netease 语料库是一个相对较小的语料库，该语料库上的实验用于验证之前

的结果。Netease 语料库中的短文本包括 8 个类别的 5465 篇单标记文本。我们采用不同的比例随机划分训练集和测试集，性能的比较如下所示：

表 5.6　Netease 语料库上不同训练测试比例下的性能比较

Train set proportion	SVM	CRFs
0.9	**0.746**	0.742
0.8	0.703	**0.740**
0.7	0.692	**0.722**
0.6	0.680	**0.706**
0.5	0.666	**0.698**
0.4	0.645	**0.683**
0.3	0.610	**0.652**
0.2	0.559	**0.614**
0.1	0.481	**0.549**

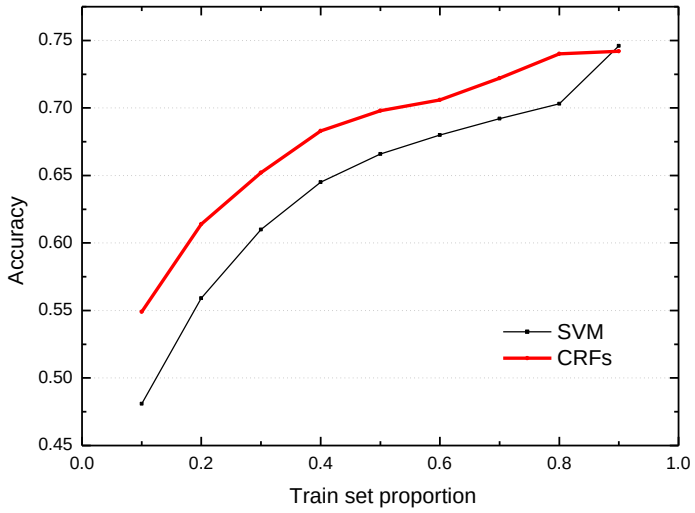

图 5.7　Netease 语料库上不同训练测试比例下的性能比较

Netease 语料库上的实验同样证明基于 CRFs 的短文本分类算法是有效的，但是 Netease 和 Sohu 语料库都是中文数据集，为了进一步评价我们的方法，我们需要把实验推广到英文数据集上。

5.4.2　英文短文本分类实验

英文短文本分类实验使用的是 Ohsumed (MEDLINE) 数据集，该数据集包含 50216 篇医学摘要。所有文本分为 23 类，标签为 C1~C23。将长摘要滤除后，我们得到了 28399 篇短文本。我们做了 4 类别分类和 5 类别分类的实验，结果如下所示：

表 5. 7 Ohsumed 语料库上的实验结果(4 分类问题)

	SVM	CRFs
C1~C4	0.560	**0.627**
C5~C8	0.457	**0.494**
C9~C12	0.501	**0.564**
C13~C16	0.590	**0.619**
C17~C20	0.432	**0.475**

表 5. 8 Ohsumed 语料库上的实验结果(5 分类问题)

	SVM	CRFs
C1~C5	0.501	**0.558**
C6~C10	0.431	**0.479**
C11~C15	0.501	**0.531**
C16~C20	0.382	**0.421**

在数据准备阶段，我们只滤除了长摘要，对数据集中的噪声文本并没有处理，所以分类的准确率相对较低，但是基于 CRFs 的短文本分类方法仍然比 SVM 分类器的准确率高。我们的方法适用于英文数据集。

5.4.3　违禁条目识别实验

查询条目分类 (query classification)广泛地使用在搜索引擎以及相关应用中，违禁条目识别就是一种特殊的查询条目分类任务。查询条目被分为两类：正常条目和违禁条目。所以，违禁条目识别本质上就是一个短文本的二分类问题。

本小节的实验使用查询日志数据集，该数据集来源于 Sogou 搜索引擎，包含 5000 个条目，其中有 1746 个条目是违禁的。

表 5.9　Query 语料库上不同训练测试比例下的性能比较

Train set proportion	SVM	CRFs
0.9	0.848	**0.888**
0.8	0.851	**0.882**
0.7	0.851	**0.882**
0.6	0.847	**0.865**
0.5	0.835	**0.850**
0.4	0.817	**0.840**
0.3	0.810	**0.823**
0.2	0.777	**0.798**
0.1	0.751	**0.776**

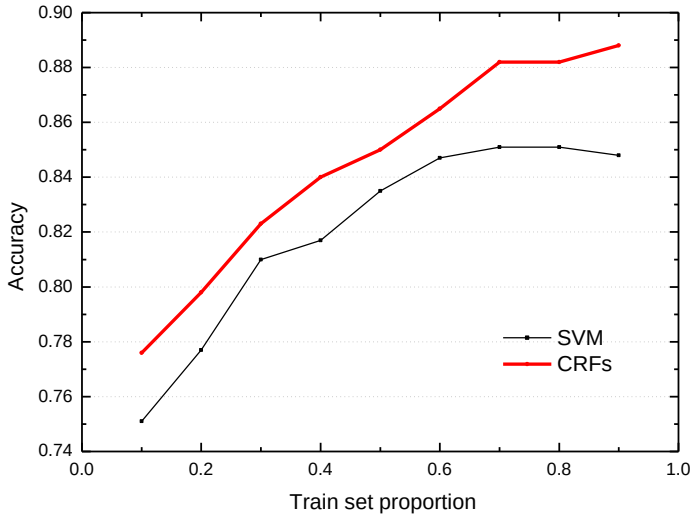

图 5.8　Query 语料库上不同训练测试比例下的性能比较

从上面的结果来看，基于 CRFs 的短文本分类方法在处理违禁条目识别问题的时候也是有效的。

综上，四个不同的数据集上的实验结果验证了我们的短文本分类算法。Sohu 和 Netease 短文本语料库是中文数据集，来源于 web 上的文本标题。Ohsumed-all 语料库是英文数据集，其来源是英文的医学摘要。我们还使用了一个查询日志集合，该集合来自 Sogou 搜索引擎。我们使用该数据集来验证基于 CRFs 的短文本分类算法在违禁条目识别中的应用。四个数据集上的结果证明我们的方法是

有效的，在所有数据集上，基于 CRFs 的短文本分类算法的性能都由于基于 SVM 的分类器。当训练集的规模很小，或是训练文本的长度非常短时，我们的方法仍然有效。

5.5　结论

短文本分类问题作为文本分类问题的一个分支，除具有共性之外，还面临一些特殊问题需要解决，因为文本长度短，特征稀疏，难以衡量短文本之间的相似性，单纯地从普通文本分类任务中移植的算法有时并不能得到很好的效果。但是，短文本分类任务在 web 环境下得到了越来越广泛的应用。如何有效地提高短文本分类准确度，已成为一个至关重要的问题。

由于短文本分类任务的特殊性，传统的文本分类方法在该领域表现不佳。因为文本较短，面临特征稀疏的问题，文本之间很少含有相同的特征，这样，文本之间的相似性不好度量，这些特殊的性质为短文本分类任务增添了极大的困难。

从直觉上来讲，短文本由于篇幅限制，涉及到的话题比较集中。所以，短文本中的特征也比较倾向与涉及同样的特征，这也被称为短文本特征的一致性。我们的分析说明特征之间的一致性随着文本的长度改变，一致性与文本长度之间是反相关关系。当文本长度非常短时，特征之间有较强的一致性。随着文章长度的增大，这种一致性会随之减弱。链式 CRFs 的结构中隐藏变量之间存在约束，该模型与短文本特征相符合，所以基于 CRFs 的短文本分类方法可以提高分类的准确度，

四个数据集上的结果证明我们的方法是有效的，在所有数据集上，基于 CRFs 的短文本分类算法的性能都由于基于 SVM 的分类器。当训练集的规模很小，或是训练文本的长度非常短时，我们的方法仍然有效。

第6章 结论

在本文的论述中，我们介绍了在基于 CRFs 的中文分词技术和短文本分类技术。由于 Web 环境的特殊性，传统的中文分词方法和短文本分类方法往往面临着一些瓶颈和缺点，本文着眼于对某些问题提出一些改进的思想，并从实验上验证了我们方法的有效性。基于 CRFs 的分词器可以达到 state-of-art 的分词效果，所以本文中的研究工作主要建立于 CRFs 分词器基础上。

首先，我们借助文本分类中的特征选择算法，定量分析中文分词任务中所使用的特征，研究证明特征选择在中文分词任务中也是适用的，可以有效地降低分词算法的时间复杂度和空间复杂度，同时不降低分词性能。

其次，我们分析了 CRFs 置信度的特性和 CRFs 分词器的局限性，CRFs 的分词器对于未登录词有比较好的切分结果，也是凭借于一些常见的特征。一旦这样的特征在句子中被省略，就很容易引入分词错误，单纯依靠 CRFs 分词器是很难解决的。另一方面，高置信度特征也可以引入干扰。PKU05 和 MSR05 上的分析工作说明了分词错误率和 CRFs 置信度之间存在反相关关系。这样，我们可以使用 CRFs 置信度作为切入点，对原分词结果进行修正，以克服 CRFs 分词器自身的弱点。不同长度的低置信区间所包含的切分错误种类不同，当低置信区间的长度较短时，如前分析，如果高置信特征对上下文产生干扰，则有可能将二元组错划为词。当低置信区间的长度较长时，则很有可能是长词被错误地切开，如成语，机构名等，对于这种情况，我们采用的修正策略就是尽可能地恢复长词。实验结果显示后处理算法可以有效提高分词的正确率，该算法在不同来源的数据集上均有效。分词的召回率和准确率同时得到提升。

第三，我们介绍了基于 CRFs 置信度和篇章上下文信息的未登录词识别算法，并使用该算法对分词结果进行后处理，以期提高分词性能。我们对于未登录词识别的算法按照如下流程：首先使用 CRFs 分词器预切分，得到每个切分的 CRFs 置信度。按照设定的阈值，选择出低置信度区间。第二步，将选择出的片段转化成后缀树，统计高频字串。第三步，对每个高频字串计算其上下文熵，若上下文熵小于设定的阈值，则对该字串进行扩充，该算法反复迭代，直到字串不再变化为止。最后得到的字串即为识别出的未登录词，利用这个信息对原

分词结果进行修正，得到新的分词结果。实验结果证明我们的算法可以有效地识别出上下文中的未登录词，提高分词结果的正确率。

另一方面，短文本分类问题作为文本分类问题的一个分支，除具有共性之外，还面临一些特殊问题需要解决，因为文本长度短，特征稀疏，难以衡量短文本之间的相似性，单纯地从普通文本分类任务中移植的算法有时并不能得到很好的效果。但是，短文本分类任务在 web 环境下得到了越来越广泛的应用。如何有效地提高短文本分类准确度，已成为一个至关重要的问题。

由于短文本分类任务的特殊性，传统的文本分类方法在该领域表现不佳。因为文本较短，面临特征稀疏的问题，文本之间很少含有相同的特征，这样，文本之间的相似性不好度量，这些特殊的性质为短文本分类任务增添了极大的困难。

从直觉上来讲，短文本由于篇幅限制，涉及到的话题比较集中。所以，短文本中的特征也比较倾向与涉及同样的特征，这也被称为短文本特征的一致性。我们的分析说明特征之间的一致性随着文本的长度改变，一致性与文本长度之间是反相关关系。当文本长度非常短时，特征之间有较强的一致性。随着文章长度的增大，这种一致性会随之减弱。链式 CRFs 的结构中隐藏变量之间存在约束，该模型与短文本特征相符合，所以基于 CRFs 的短文本分类方法可以提高分类的准确度，

四个数据集上的结果证明我们的方法是有效的，在所有数据集上，基于 CRFs 的短文本分类算法的性能都由于基于 SVM 的分类器。当训练集的规模很小，或是训练文本的长度非常短时，我们的方法仍然有效。实验证明，将短文本分类问题转化成序列标注问题，用 CRFs 解决该问题，可以得到更高的分类准确度。

参考文献

[1] Fuchun Peng, Fangfang Feng, and Andrew McCallum.2004. Chinese segmentation and new word detection using conditional random fields. In Proceedings of COLING 2004, pages 562-568, Switzerland.

[2] M.S. Sun, D.Y. Shen and B. K. Tsou. 1998. Chineseword segmentation without using lexicon and handcrafted training data. In Proceeding of COLINGACL'98, pages 1265-1271, Quebec, Canada.

[3] Xue, Nianwen. 2003. Chinese word segmentation as character tagging. In Journal of Computational Linguistics and Chinese Language Processing, 8(1).

[4] Joachims T. Text categorization with Support Vector Machines: Learning with many relevant features. Machine Learning: ECML-98, Tenth European Conference on Machine Learning: 137-142, 1998.

[5] McCallum A, Nigam K. A Comparison of Event Models for Naïve Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization, 1998.

[6] Yang Y M and Liu X, A re-examination of text categorization methods, In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval: 42-49, 1999.

[7] Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47, 2002.

[8] David D. Palmer. 1997. A Trainable Rule-based Algorithm for Word Segmentation. In The Proceedings of ACL 1997, pages 321-328.Madrid, Spain.

[9] K.S. Cheng, G.H. Young, K.F.Wong. 1999. A study on word-based and integral-bit Chinese text compression algorithms Jounal of the American Society for Information Science, Vol. 50(3):218-228.

[10] R. Sproat et al.. 1996. A stochastic finite-state wordsegmentation algorithm for Chinese. Computational Linguistics. 22(3): 377-404.

[11] Dai Yubin, Teck Ee Loh, Christopher S. G. Khoo. 1999. A new statistical formula for Chinese text segmentation incorporating contextual information. In Proceedings of ACM SIGIR, pages 82-89. CA, USA.

[12] Teahan W.J., Y. Wen, R. McNab, and I.H. Witten. 2000. A Compression-based Algorithm for Chinese word segmentation. Computational Linguistics, 26(3):375-393.

[13] Ruiqiang Zhang, Genichiro Kikui and Eiichiro Sumita. 2006. Subword-based Tagging for Confidence dependent Chinese Word Segmentaion. In Proceedings of the COLING/ACL, Main Conference Poster Sessions, pages 961-968. Sydney, Australia.

[14] Vladimir N. Vapnik 著, 张学工 译. 统计学习理论的本质. 清华大学出版社,2004.

[15] Adam L. Berger and Stephen A. Della Pietra and Vincent J. Della Pietra. 1996. A Maximum Entropy approach to Natural Language Processing. Journal of Computational Linguistic, 22(1): 39-71.

[16] McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. Proc. ICML 2000 (pp. 591–598). Stanford, California.

[17] John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In Proceedings of ICML-2001, pages 591–598.

[18] Duda R, Hart P, Stork D. Pattern Classification. Wiley-Interscience, 2000.

[19] Mitchell T. Machine Learning. WCB/McGraw-Hill, 1997.

[20] Bishop C. Pattern Recognition and Machine Learning. Springer, 2006.

[21] Salton G, McGill M. Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, 1986.

[22] Baeza-Yates R A, Ribeiro-Neto B A. Modern Information Retrieval. ACM Press/ Addison Wesley, 1999.

[23] Joachims T. Text categorization with Support Vector Machines: Learning with many relevant features. Machine Learning: ECML-98, Tenth European Conference on Machine Learning: 137-142, 1998.

[24] Richard Sproat and Tom Emerson. 2003. The first international chinese word segmentation bakeoff. In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, July.

[25] T. Emerson. 2005. The second international Chinese word segmentation bakeoff. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 123-133.

[26] G. Levow. 2006. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. In Proceedings of SIGHAN-2006, 108-117. Sydney, Australia.

[27] Changning Huang and Hai Zhao. 2005. Chinese Word Segmentation: A Decade Review. Journal of Chinese Information Processing, 2007, Vol. 21(3): 8-20.

[28] Salton G, Wong A, Yang C. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11): 613-620, 1975.

[29] Lewis D. Evaluating text categorization. Proceedings of Speech and Natural Language Workshop: 312–318, 1991.

[30] Yang, Y M, Pedersen, J O. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of 14th International Conference on Machine Learning: 412–420, 1997.

[31] Zelikovitz, S. and Hirsh, H., “Improving short-text classification using unlabeled background knowledge to assess document similarity”, Proceedings of the Seventeenth International Conference on Machine Learning, 2000.

[32] Shen, D., Pan, R., Sun, J.-T., Pan, J. J.;Wu, K., Yin, J., and Yang, Q, “Query enrichment for web-query classification”, ACM Trans. Inf. Syst. 24(3), 2006, pp.320–352.

[33] D. Bollegala, Y. Matsuo, and M. Ishizuka, “Measuring semantic similarity between words using Web search engines”, Proc. WWW , 2007.

[34] M. Sahami and T. Heilman, “A Webbased kernel function for measuring the similarity of short text snippets”, Proc. WWW, 2006.

[35] W. Yih and C. Meek, “Improving similarity measures for short segments of text”, Proc. AAAI, 2007.

[36] JS. Banerjee, K. Ramanathan, and A. Gupta, “Clustering short texts using Wikipedia”, Proc. ACM SIGIR, 2007.

[37] X. Phan, L. Nguyen, and S. Horiguchi, “Learning to Classify Short and Sparse Text and Web with Hidden Topics from Large-scale Data Collections”, The International World Wide Web Conference Committee (IW3C2), 2008.

[38] P. Schonhofen, “Identifying document topics using the Wikipedia category network”, Proc. the IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[39] Jan Blaˇt´ak, Eva Mr´akov´a and Luboˇs Popel´ınsk´, “Fragments and Text Categorization”, Proceedings of the ACL, 2004.

[40] Jingyang Li and Maosong Sun, “Scalable term selection for text categorization”, In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2007, pp. 774-782.

[41] Jingyang Li, Maosong Sun, and Xian Zhang, 2006. “A comparison and semi-quantitative analysis of words and character-bigrams as features in Chinese

text categorization”, In Proceedings of COLING-ACL, Association for Computational Linguistics , 2006, pp.545–552.

致　谢

衷心感谢导师孙茂松教授对本人的精心指导。在本文研究工作期间，无论是选题阶段、中期进展阶段，还是最后学位论文的撰写，孙老师都给予了我极大的支持和鼓励。他的言传身教将使我终生受益。

在研究期间，我得到了实验室许多同学的关心和帮助。在此向李景阳，李伟，乔维，张燕，司献策，郑亚斌，刘知远，蒋琪夏，张开旭等同学表示衷心的感谢！同时也感谢我的父母，是你们在生活上帮助我，精神上支持我，正是在你们的鼓励下，我才能走完这些年的人生历程。

本研究得到清华-波音公司国际合作项目“Robust Chinese Word Segmentation and High Performance English-Chinese Bilingual Text Alignment”和 863 项目“大规模网络图文数据的语义分类和适度理解技术研究”（2007AA01Z148）的支持。波音公司的薛平博士就中文分词部分同我进行了认真的讨论，从中我得到了许多有益的建议，在此一并致谢。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1985 年 1 月 27 日出生于陕西省西安市。

2003 年 9 月考入清华大学计算机科学与技术系计算机科学与技术专业，2007 年 7 月本科毕业并获得工学学士学位。

2007 年 9 月免试进入清华大学计算机科学与技术系计算机应用专业攻读硕士学位至今，师从孙茂松教授。

发表的学术论文

[1] Yabin Zheng, Shaohua Teng, Zhiyuan Liu, Maosong Sun. Text Classification Based on Transfer Learning and Self-Training. ICNC-FSKD 2008. (EI 检索号：085111800412)

[2] Yabin Zheng, Zhiyuan Liu, Shaohua Teng and Maosong Sun, Efficient Text Classification Using Term Projection, AIRS 2009.

[3] Shaohua Teng, Maosong Sun, Yabin Zheng. Using Conditional Random Fields for Very Short Text Classification, submitted to ALPIT 2009.

[4] Qixia Jiang, Xiance Si, Shaohua Teng, Maosong Sun. Particle Mixed Membership Stochastic Block Model on Valued Graphs, submitted to CIKM 2009.