密级：保密期限：

北京郵電大學

硕士学位论文

题目：基于分块的汉语句法分析技术设计与应用

学　　号：2013140363

姓　　名：何德铸

专　　业：计算机技术

导　　师：王小捷

学　　院：计算机学院

2015 年 12 月 25 日

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在年解密后适用本授权书。

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

基于分块的汉语句法分析技术设计与应用

摘　要

句法分析作为自然语言处理核心的技术之一，是许多自然语言处理任务的基础。对于一些需要利用深层句子结构信息的语言处理任务而言尤其重要。因此，句法分析技术得到了广泛的研究。但是，由于句法分析的固有难度，与其他自然语言处理基础技术相比，其水平目前仍相对较低，尤其是对于汉语这种句法结构相对自由的语言，还有许多问题需要进一步深入研究。

本文针对汉语语言，在综合分析当前若干句法分析模型特点的基础上，开展了如下的工作。

设计并实现了一种基于分块的汉语句法分析技术。该技术将整句句法分析分为三个阶段：首先将句子进行分块，然后分别在各块中进行块句法分析，最后将各块分析得到的句法树合并得到整句句法分析结果。在分块阶段，通过将分块任务转化成序列标注任务，实现了一个基于条件随机场模型的分块算法。在块句法分析阶段，为了降低问题复杂度，进一步将其分解为块结构分析和标签分类两个子阶段。其中，将块结构分析转化为序列标注问题，实现了一个基于条件随机场模型的块结构分析算法，并引入集束搜索技术来缓解错误累加问题；综合利用多种特征实现了基于条件随机场模型的标签分类子算法。实验结果表明了本文所提出的汉语句法分析技术的有效性。

最后，将所提出的句法分析技术应用于信息检索任务中，实现了一个面向旅游领域的信息检索系统，实验表明，加入句法分析技术有效提升了信息检索系统的性能。

关键词　汉语句法分析　分块　条件随机场　集束搜索

CHUNK BASED CHINESE SYNTACTIC PARSING AND ITS APPLICATION

ABSTRACT

As one of the most important technologies of natural language processing, syntactic parsing is the foundation of many natural language processing tasks. It is particularly important for language processing tasks that require the use of deep sentence structure. Therefore, syntactic parsing technology has been widely studied. However, there still need to be more promotion for this technologies than others due to its difficulty. We should do further work on it, let alone the uniqueness of its much more free syntactic structure.

In this paper, we carry out some work based on the characteristics of several former work of syntactic parsing.

We design a chunk based Chinese syntactic parsing in this paper. We divide into three stages. First, sentences are divided into chunks. Then we get the chunks parsed. Finally we merge all the parsed chunks into a whole syntax tree. During chunking stage, a conditional random field model is implemented by taking chunking as a sequence labeling task. In the chunks syntactic parsing stage, it is divided into two sub module which consists of chunks structure analysis and label classification in order to reduce the complexity. We implemented a chunks structure analysis algorithm based on conditional random field model by transforming it into a sequence labeling problem. And beam search is used to alleviate the error accumulation problem when parsing. We make a classifier for syntactic tag classification based on conditional random field model by using rich features. The experimental results show the effectiveness of this Chinese syntactic parsing addressed by this paper.

Finally, we apply this syntactic parsing to an information retrieval

system based on tourism.It shows that and this syntactic parsing just bring a lot of improvement to the system.

KEY WORDS：Chinese syntactic parsing; chunking; conditional random field; beam search

目录

第一章 绪论

1.1 背景及意义

句法分析是语言分析的一个基础性任务，是对句子进行深层分析的一个主要途径。句法分析技术是自然语言处理的核心技术之一，在许多自然语言处理应用任务中都起着重要作用。

句法分析的目的是识别出句子所包含的句法成分以及这些成分之间的句法关系。句法分析的结果一般以树结构表示，即句法树。句法树中每个结点都有相应的标记，其中叶子结点的标记为句子中的词本身，叶子结点的父结点为词的句法范畴(词性)，其他非叶子结点的标记为该结点的子结点所构成的短语结构。图 1-1 给出了句子“*全面推行教育收费公示制度。*”的句法树。

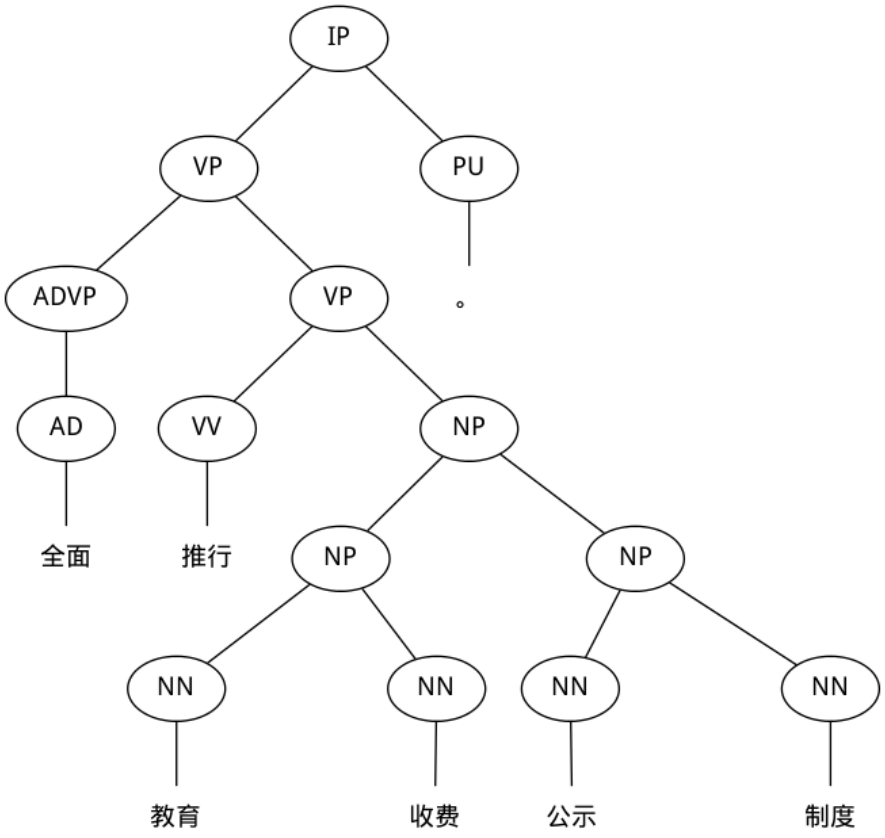

图 1-1“全面推行教育收费公示制度。”的句法分析结果

在图 1-1 中，“*教育*”和“*收费*”的词性均为名词(NN)，这两个 NN 组成了上一级的结构名词短语(NP)，同样，“*公示*”和“*制度*”这两个 NN 也构成一个 NP，进一步，这两个 NP 又共同构成一个更大范围的新的 NP。句子其他部分也按此方式结合，最终构成了一颗完整的句法树。

可以看到，句子的句法树结构比线性的句子表示提供了更为丰富的语言信息，对很多自然语言处理应用而言这些句法信息可以有效地提高应用的性能。

在信息检索中，除了可以利用词汇和词性信息，还可以利用句法分析提取出适当的句法成分可以使检索系统更加精确，泛化能力更强。例如：在检索“*请问下在早晨 5 点从五道口到天安门要怎么坐地铁去呢？*”时，一般的方法在提取关键词时可能会将“*问下*”作为关键动词，从而造成信息检索的错误。加入句法分析信息后，我们能了解到“*请问下*”后跟随的短语结构是整个句子关键结构，从而轻松地提取到句子的主要实体（根据词语所在句法结点的覆盖度和深度判断）为“*五道口*”与“*天安门*”，且关键动词为“*去*”，关键名词为“*地铁*”，关键时间词为“*5 点*”。

在机器翻译中，经常使用转换模型来改变输入语言的结构而使之符合目标语言的语法规则。转换模型的第一步需要进行句法转换，来对两种语言的词和短语进行一一对应，即语料对齐。通过句法转换能更好的学习两种语言在结构上的变换策略，使机器翻译的结果更加接近翻译的目标语言的结构[1]。例如在翻译“*there was an old man gardening*”时，直接逐词翻译得到“*有一个老人园艺*”。通过句法分析我们可以得到其主要成分有“*there*”、“*was*”、“*an old man*”和“*gardening*”，接着利用句法结构和词汇转换，用中文代替英语单词，得到的最后翻译结果为“*这里有位正在做园艺的老人*”。

在情感分析的评价对象提取任务中，利用句法分析也可以有效地提高抽取的准确性。例如，对于句子“*这款电脑的续航能力很强，但是价格太高。*”不使用句法信息时一般只能识别出“*能力*”和“*价格*”作为评价对象。而引入句法分析后，得到的评价对象会是 “*续航*”、“*能力*”、“*续航能力*”和“*价格*”，虽然同时也引入了噪声，但是，句法分析可以帮助避免了错过正确的评价对象。

句法分析的应用范围还在不断扩展，随着计算能力的增强，也使得在更多应用中部署句法分析技术成为可能。由于这种应用需求的不断增强，人们对句法分析的投入也逐渐增大，产生了不少的研究成果，但是，由于句法分析本身的难度，尤其是在汉语中，语言结构相对比较灵活，因此，相比于汉语切分、词性标注、命名实体识别等技术，汉语句法分析技术的水平亟待提高。

1.2 句法分析的研究现状

1.2.1 句法分析的发展历史

早期自然语言句法分析的研究主要是针对英语来展开的，在每个时间段内最先进的句法分析算法和模型通常都会先在英语上进行实验，然后再为其他语言所借鉴。所以可以从英语句法分析的发展来了解句法分析技术的发展历程。

句法分析的工作开始于上世纪 50 年代，1949 年美国洛克菲勒基金会的 Weaver 最早提出了统计机器翻译思想，1954 年美国乔治敦大学在 IBM 公司的帮助下开展了最早的以简单“查字典”为基础的机器翻译系统的实验。该实验最终最终没有取得成功，但是人们从中意识到在机器翻译任务中，他们必须找到一种更深层次的方法来表达句子，自然语言的句法分析研究从此拉开序幕。

句法分析方法可以简单地分为基于规则的方法和基于统计的方法两大类。

基于规则的句法分析方法在早期占据统治的地位。基于规则的方法强调语言学家对语言现象的认识，全凭语言学家制定的规则集，不断处理文本分析的错误，制定、修改新的规则，直至系统性能满足需求。

50 年代末期，Chomsky 构建了上下文无关语法的理论体系，许多系统都是利用该思想搭建而成。如 Robert Lindsay 于 1963 年开发的 SAD-SAM 系统利用 Chomsky 的转换语法和形式化理论构建了一个句法分析器。这个系统只是利用了大约 1700 个基本的英语单词和简单的上下文无关的英语文法，从左至右的对输入句子进行剖析并生成推导树，然后利用推导树进行相关的自然语言处理任务。很明显 SAD-SAM 存在词汇量小、语法覆盖窄的问题。

60 年代中后期，许多基于关键词分析和模式匹配的系统得到广泛应用。比如 SIR、STUDENT、ELIZA 系统，这些系统通过将输入文本与事先优选后的模板进行匹配。其中最有名的是由 MIT 在 1966 年编写的 ELIZA 系统，ELIZA 的主要操作是左匹配用户输入的句子，然后根据合适的右规则生成回答。该系统只是通过少数几个关键词来进行索引匹配模板，所以它只能解决某些特定的问题，其语言理解能力还是比较肤浅。

70 年代初期，美国人工智能专家 W.A.伍兹提出了扩充转移网络（Augmented Transition Network, ATN），该文法在增加正则表达式的能力的同时，还克服了有限状态机在表达上下文无关文法时存在的限制，其可以生成所有的递归可枚举的语法规则。但是 ATN 有着移植性较差，性能差的缺点，且当句子的前面部分未知情况下，句子的句法分析结果会较差。

90 年代，仍有一些基于规则的句法分析系统，比如 1991 年开始的 NLPwin

是微软研究院开发的一套旨在提供 Windows 平台上的自然语言处理工具。虽然 NLPwin 是一个基于规则的系统，但是它却能够覆盖多达 7 种语言的自然语言处理，其中包括英文、中文。NLPwin 可以从标注的句子集中学习到语法规则并利用丰富的在线词汇和语义资源。它使用二元增强短语结构文法（augmented phrase structure grammar，APSG），所以在结构的生成过程中能保持二元形式[3]。

基于规则的句法分析方法的缺点是：在处理大规模真实语料时，很可能会出现系统可维护性和可迁移性差、语法规则覆盖度有限等缺陷。

80 年代末，大量科研机构开始构建大规模标注语料库，从而促进了基于统计学习模型的句法分析方法的蓬勃发展。与传统的基于规则的句法分析方法相比，基于统计的方法更加强调从真实的句子中获取知识，而非依靠语言学家的直觉。早期的基于统计的句法分析模型大多只是简单地基于概率上下文无关文法（Probabilistic Context Free Grammar，PCFG），这种方法结果很不理想，一个主要原因是因为 PCFG 在句法分析过程中忽略了词汇语义信息。在这之后的一段时间内，句法分析的研究的工作基本都是在探讨如何在 PCFG 模型的基础上，使 PCFG 模型能够表达更加丰富的信息，从而提高句法分析的性能。

Magerman[4][5]于 1995 年在他提出的基于决策树的 PCFG 句法分析模型中率先引入词汇信息。其中心思想是把结构树的建立转化为一个自底向上的决策问题，每一步决策的概率由当时结构树的上下文信息来确定。他利用 Penn Treebank 语料进行模型的训练和测试，准确率和召回率分别达到了 84.5%和 84.0%。此后，Penn Treebank 成为了句法分析进行训练测试的公共平台，模型或者算法利用这个树库来进行评测和比较。

Ratnaparkhi[6]在 1999 年设计了一个移进归约式的模型，利用最大熵方法来进行参数估计，该模型将句法分析任务分成三个子任务：词性标注、句法块划分（Chunking）和建树过程，并在搜索的时候采用集束搜索(Beam Search)技术，分析结果的准确率和召回率分别达到了 87.5%和 86.3%。

Collins[7]于 1999 年提出的基于中心驱动的句法分析模型是目前影响力最大的句法分析模型之一。该模型利用由语言学家手工制定的句法树中心成分规则标注训练语料中的每个句法子树的中心成分，并在训练模型的过程中统计中心成分与修饰成分、中心词与修饰词之间的概率，最后利用这些概率选取最优的句法分析树。这种方法有效地缓解了 PCFG 模型的数据稀疏问题，提高了模型的表示能力，使句法分析的性能得到显著的提升，其分析结构的准确率和召回率分别是 88.3%和 88.1%。这一模型引发一系列对于中心驱动模型改进的热潮，句法分析得到很大的发展。

Collins[8]进一步于 2000 年在其 1999 工作的基础上引入重排序算法进行句法

分析。由于句法分析的解码空间非常大，在建树的过程中只用到了一些局部的特征，但是很多非局部特征对选择正确的句法树也有很大的作用。重排序的方法首先利用一个已有的模型输出评价最佳的 k 棵树及其得分，然后基于非局部特征利用重排序算法对这 k 棵树进行重新打分，最后利用两次的打分评选出最优的句法树。该模型的准确率和召回率分别达到了 89.9%和 89.6%。

Klein 和 Manning[9]在 2003 年提出了一个非词汇化的句法分析模型。该模型利用语言学知识和统计结果对句法树的不同情况手工地进行标记的分割，标记的对象包括父结点、词性标注分割、单叉树内外部结点标记等，这些分割赋予了句法树标记以更多的意义。该方法实现的模型非常简单、易于理解、易于实现，且最终的 F1 值达到了 86.32%，因此获得了 2003 年 ACL 大会的最佳论文奖。

Henderson[10]于 2003 年提出了一个基于神经网络的统计句法分析模型。模型利用神经网络模型来估计左角分析法中生成模型的参数，并用这些参数搜索概率最大的句法分析过程。模型的 F1 值达到 88.8%，只低于当时最优的结果不到 1%，说明了神经网络结构也能很好的表示无限的句法结构，开创了基于神经网络进行句法分析的先河。

Petrov[11]在 2006 年综合 Matsuzaki[12]和 Klein&Manning[9]模型的优点，构建了一个基于分割和合并句法标记的策略自动标注系统，并利用参数平滑算法，提高系统的鲁棒性。Petrov 的模型简单、准确且具有很高的解释性，由此构建的 Berkeley Parser 在 Penn Treebank 语料上取得了 90.2%的 F1 值，超过当时所有的词汇化句法分析模型。

Shindo[13]于 2012 年提出了符号树替换文法（Symbol-Refined Tree Substitution Grammars，SR-TSGs）来进行句法分析。SR-TSG 是传统 TSG 模型的一个扩展，区别是它可以改变每个非终结符来适应训练数据。该模型构建了一个全新的概率 SR-TSG 模型来简化 CFG 的规则，并且使用基于马尔科夫链的蒙特卡罗（Markov Chain Monte Carlo，MCMC）抽样进行训练。该句法分析模型单一系统的 F1 值达到 91.1%，在增加重排序之后的 F1 值达到了 92.4%，这是目前基于 Peen Treebank 英文树库的句法分析得到的最佳结果。

Hall[14]在 2014 年通过直接从表层提取有效特征而不是通过语法结构标记信息来构造句法分析器。该模型利用条件随机场模型（Random Conditional Fields，CRF）来训练句法分析模型，只利用少量的特征模板和表层特征取得了可以和 Berkeley 句法分析相媲美的结果，这种方法几乎不需要研究员考虑语法的问题，将句法分析的主要难点转化成特征工程，大大降低的句法分析算法的复杂度。该方法的 F1 值达到了 89.2%。

上述句法分析模型的基本思路是是通过给更合理的句法树打尽可能高分的

方式挑出好的句法树，为此需要预先把所有可能的候选句法树都找出来。这一功能通常由句法分析模型中的句法分析算法来实现，因此句法分析算法是句法分析模型的基础，实现句法分析算法的程序通常被人们称为解码器（Decoder）。

句法分析算法大致分为两类：基于表的句法分析算法以及基于下推自动机原理的句法分析算法。基于下推自动机原理的算法具有线性的空间复杂度和指数级时间复杂度的特点，典型的代表是 LR(首次出现的英语缩写需要写全拼，后面同)分析算法[15]和 GLR 算法[16]。基于表的分析算法具有句长(句子中包含的词的数量)二次方的空间复杂度和句长三次方的时间复杂度的特点。基于表的分析算法还细分为自底向上和自顶向下两种，由 Cocke、Kasami 和 Younger 提出的 CKY 算法[17][18]，是定型的自底向上算法。CKY 算法实现了 Viterbi 算法来进行搜索空间的降低，其限制是只能处理 Chomsky 范式的规则，即二元规则，所以处理上下文无关文法的时候需要将语法规则转化成 Chomsky 范式。Earley[19]提出的 Earley 算法是典型的自顶向下的分析算法，该算法的核心是从左到右的传递，利用动态规划的方法，填充一个线图（Chart）数组，这个线图保存着一个状态表来表示已经生成的部分句法树。

基于统计的句法分析模型需要树库来获取模型的语法规则和模型参数。近几年，大规模树库的建设大大推进了基于统计的句法分析技术的发展，为句法分析技术迎来新的大发展打下了坚实的基础。

英语的树库发展较早，英国 Lancaster 大学的 UCREL 研究小组[20]在 1984 到 1988 年五年间累计标注了 200 多万词的树库语料。美国 Pennsylvania 大学标注的宾州树库（Peen Treebank）[21]是目前句法分析研究者用的最多的树库。宾州树库是 ATIS 和华尔街日报（Wall Street Journal，WSJ）树库的发展，从 1991 年第一次出现到现在，经过二十多年的发展，宾州树库不断地完善和扩充树库，已经成为了目前英语句法分析所公认的标准训练集合测试集。除此之外，还有 IBM 的计算机手册的树库和美国图书馆树库。

1.2.2 汉语句法分析的现状

相比于英语句法分析技术，汉语的句法分析技术起步较慢，一个重要的原因是汉语树库建设的滞后，同时也存在一些与英语相比固有的困难。汉语句法结构与英语存在很多差异，其中最明显的表层差异是汉语词之间没有空格作为词的分界符，因此在句法分析前必须先进行汉语特有的分词（Word Segmentation）。其次，汉语重语义，英语重结构，这些特点大大加大了汉语句法分析的难度，表现在同时期的句法分析精度上，汉语都会低于英语句法分析的水平。

即使面对种种难点，汉语句法分析在国内外学者的工作和努力下还是取得了不少进展。

Bikel&Chiang[22]于 2000 年第一次将 Collins[7]基于中心驱动的句法分析模型应用到了当时刚发布的宾州中文树库（Penn Chinese Treebank，CTB），并根据英语与汉语的区别改修改了中心词规则，取得了 77%的准确率和 78%的召回率。

周强[23]在 1999 年提出了利用局部优先信息对汉语句法分析算法进行改进的新方法。该方法通过去除一些局部优先组合能力较小的句法成分，从而提高整体的句法分析效率。

张玥杰[24]在 2000 年实现了面向数据的句法分析模型（Data-Oriented Parsing，DOP）。该方法首先利用带有句法标记的句法分析树库自动构建一个知识源，取得待分析的句子的所有合法片段组合，然后将句子与初选结果进行基于相似的概率评估，进而完成句子的组合分析过程。

Luo[25]在 2003 年提出了基于字的句法分析模型。汉语的分词精确度会影响句法分析的效果，所以该模型将树库中的词和词性转换成字和字词性，然后在统一的框架下利用最大熵模型同时进行分词、词性标注和句法分析。

吕雅娟[26]在 2003 年针对汉语句法语料缺乏的情况，提出了一种自动获取汉语句法知识的方法。该方法用双语对齐技术将可信度较高的英语句法分析语料映射成汉语句法树，并从中学习到汉语句法知识。

曹海龙[27]在 2006 年率先在 CTB5.0 上应用中心驱动模型，验证了词汇化统计模型对汉语的可行性，并提出一个两级中文句法分析方法，该方法提出分治策略，即区别对待基本短语和复杂短语。两级中文句法分析方法显著地提高了句法分析的准确率和召回率，分别达到了 87.5%和 87.95%。

李军辉[28]在 2006 年实现了一种基于循序渐进方式的汉语句法分析。该方法在分析过程中优先识别出容易识别的组块，然后利用这些组块提供更加丰富的上下文信息进行较为复杂的组块识别，并在 CTB 上取得了 83.07%（<=40 words）的 F1 值。

Wang[29]在 2006 年将移进规约（Shift Reduce）的决策模型应用在句法分析上，使句法分析的解码算法降低到线性复杂度，极大提高了句法分析的解码速度和准确率。该模型的准确率和召回率达到了 88.3%和 88.1%，并比当时最好的句法分析模型速度快了 40-270 倍。

袁里驰[30]在 2011 年提出基于词聚类的依存句法分析，利用邻接词和语义依存关系计算相似度，并进行聚类，然后将句法分析中出现的未登录词(Out of Vocabulary, OOV)转换成该词的聚类，大大缓解的数据稀疏问题。

Qian[31]在 2012 年提出基于字符的分词、词性标注、句法分析联合模型，将

三种自然语言处理技术联合，并利用分词模型丰富的特征改善其他两种技术的性能，最终的句法分析取得较好的性能。

二十世纪 90 年代，汉语的语料库发展迅速，涌现许多优秀的语料资源。

北京大学计算语言学研究所与日本富士通公司合作，将 1998 年的人民日报语料加工成 2700 万字的汉语语料库[32]，其中包含分词、词性标注，专有名词标注。人民日报标注语料大大加快汉语自然语言处理的发展，特别地对中文分词任务的推动作用尤为显著。

2000 年，语言数据协会（The Association for Computational Linguistics，ACL）发行了宾州中文树库[33]，该树库属于短语结构树库，沿用英语 Peen Treebank 的标注体系对汉语句子进行标注。最初的 CTB1.0 只有 100000 词，4185 句，截止 2013 年，CTB 已经发展到了 CTB8.0 版本，包含 3007 个文本文件，71369 句，1620561 词，2589848 个字，包含网络采集语料、通讯新闻、新闻杂志、广播新闻、广播谈话、博客等类型语料。目前 CTB 已经成为了国际上最具影响力的中文树库之一。

1998 年，清华大学计算机系的周强和黄昌宁等人构建了国内第一个大规模汉语树库——清华大学汉语树库（Tsinghua Chinese Treebank，TCT）[34]。TCT 选择了文学、学术、新闻和应用四大体裁的语料进行加工标注，对汉语的各种复杂的语言现象进行了标注实践，完成了 100 万字的规模的汉语树库，是当时世界上规模最大、信息标注最丰富的汉语句法树库。

哈尔滨工业大学机器智能与翻译研究室建设的哈工大汉语树库[35]也是一个著名的汉语树库。哈工大汉语树库共 17283 句，数据的每个句子都是经过挑选，使得树库具有句型和领域多样性。该树库参考了宾州 CTB 树库、清华 TCT 树库以及台湾中央研究院的中文句法树库的标签体系。其包含 42 个不同的词性标签以及 10 个标点标签，20 个短语标签。

哈尔滨工业大学信息检索研究中心于建立的 1 万句的依存句法树库（HIT-IR-CDT）[36]是国内最具代表性的依存语法树库之一。该依存树库建立于 2005 年，第一版树库使用的数据是 1998 年上半年《人民日报》，树库的标注标签包括 28 种不同的词性和 24 种依存标签。

这些语言资源，尤其是富含句法信息的语言资源对于推动汉语句法分析技术的发展产生了积极的作用。

1.3 论文的主要内容概述和章节简介

本文在前人的工作上，利用丰富的特征将句法分析转化成序列标注问题。本文利用条件随机场（Conditional Random Field，CRF）建立序列标注模型来进行句法分析的研究。主要的研究内容如下：

（1）句子分块。研究了基于 CRF 模型的分块算法。首先对训练数据进行预处理，利用介词、连词、标点符号作为句子分割符将训练语料分割成块，确保块内没有这些分割标记，而后在此数据上训练基于词特征的条件随机场模型。

（2）块内与块间结构分析。借鉴移进规约模型将结构分析问题转化成序列标注问题，研究利用丰富的词特征、结点结构特征、中心词特征以及句法特征构建基于 CRF 序列标注模型的结构分析模型，并讨论了不同标注标签体系的可行性。

（3）新结点标注。块内与块间结构分析时只是形成了空的新结点，需要进一步为结点打上句法标注信息。系统利用分类模型来构建新结点标注模型，并探讨了不同分类模型和各种特征对标注性能的影响。

（4）集束搜索。1-Best 的句法分析模型很容易在句法分析的早期错过正确的句法分析树，导致错误蔓延，不断放大分析错误规模，极大降低句法分析的性能。本文采用集束搜索技术，每次保留最佳的 K 棵树，并讨论了不同打分算法对系统性能的影响。

本文各章节安排如下：

第一章主要概述了本文的研究背景及意义、句法分析的研究现状和发展历史，并描述了本文的主要研究内容以及全文的章节安排。

第二章简要介绍了本文研究所需要的基础知识，包括中心驱动模型的汉语句法分析、层次化汉语长句句法分析方法和移进规约句法分析模型，最后介绍了条件随机场模型。

第三章描述了基于句子分块的汉语句法分析技术的设计。主要包括语料的预处理、句子分块的设计、句法分析模块的设计以及集束搜索模块的设计。

第四章描述了基于旅游领域的信息检索系统，并讨论了句法分析对该信息检索系统性能的影响。

第五章是对全文工作的总结，并根据在实现本文系统的过程中发现的一些问题提出句法分析技术的未来研究的展望。

第二章 基础知识

本章将介绍本文在研究中相关的句法分析知识和机器学习模型。首先介绍上下文无关文法与基于概率的上下文无关文法，然后描述基于中心驱动模型的汉语句法分析模型以及其他句法分析模型，最后简要介绍本文系统的主要机器学习模型——条件随机场模型。

2.1 概率上下文无关文法

2.1.1 上下文无关文法

乔姆斯基（Noam Chomsky）曾经把语言定义为：按照一定规律构成的句子和符号串的有限或无限的集合。我们称这个规律为文法。文法如同是句法分析的算法阶梯，决定着句法分析模型的设计。在乔姆斯基的语法理论中，文法被分为四种类型：0 型文法、1 型文法、2 型文法和 3 型文法，分别称为无约束文法（Unrestricted Grammar）、上下文有关文法（Context Sensitive Grammar，CSG）、上下文无关文法（Context Free Grammar，CFG）和正则文法（Regular Grammar）。这四个文法的对文法规则的限制依次严格，所以描述语言的能力依次变弱。其中，上下文无关文法具有形式简单，多项式时间的分析效率的特点，而被广泛地应用在自然语言处理的句法分析任务中。

上下文无关文法定义是：

上下文无关文法为一个四元组：

$$G = (N, \sum, P, S)$$

其中，N 是非终结符的有限集合。

Σ 是终结符号的有限集合，且$N \cap \sum = \emptyset$。

P 是文法规则的有限集合，$P = \{\alpha \to \beta\}$，其中$\alpha, \beta \in (N \cup \sum)^*$，但是α至少含有一个非终结符号。

S 为句子符或初始符，$S \in N$。

虽然上下文无关文法可以描述和生成自然语言，但是由于其文法规则的右端可以是任意长度的符号，导致其描述和生成自然语言的过程十分复杂。为此乔姆斯基提出了一种变种的上下文无关文法——乔姆斯基范式(Chomsky normal

form)[37]。该文法将上下文无关文法的文法规则改写成 A→BC 或 A→β 的形式，其中， $A，B，C \in N，\beta \in \sum$。具有这样的改写规则的上下文无关文法，它的推导树均可化为二元形式。利用乔姆斯基范式，我们只需用二分法就可以分析自然语言，极大地降低了分析的复杂度。

很多自然语言都具有二分的特性，比如英语和汉语在句法结构上，一般都是二分的。在语言学史上，不少语言学家在描写自然语言的工作中，已经发现了自然语言的这种二分特性。因此乔姆斯基范式的提出也是契合了自然语言的特性。目前主流的句法分析算法都是基于乔姆斯基范式设计的，例如 CKY 算法。

上下文无关文法用以表示自然语言的过程中，非终结符号集 N 即句子中的词的词性和短语标记，终结符号集∑代表句子中的词，而规则集合 P 则是每个短语标记推导出子结点标记的规则或词性推导出词的规则。比如：图 1-1 中，结点标记“IP”、“VP”等和词性标记“PU”、“AD”和“VV”等都是非终结符号，词“全面”、“推行”等都是终结符号，而从中可以提取到的句法规则有：

IP -> VP PU

VP -> ADVP VP

ADVP -> AD

AD ->全面

…

其中，IP 表示简单句，VP 表示动词短语，PU 表示标点符号词性，其他符号的意义请参考附录 1 与附录 2。

2.1.2 基于概率的上下文无关文法

由于句法歧义的原因，对一个确定的句子的推导树的规模是指数级的。具体，针对一个具有 N 个词的句子，假设都是二分结构，则单单是结构不同的推导树的规模就达到了C_{2n-2}^{n-1}/n，这个数量级的增加的可怕可以从附录 3 中窥得，只是 20 词的句子其规模达到了超过 10 亿数量级。如果考虑到推导树结点的标记，且非终结符标记共 K 个，则其规模达到$O(k^n * C_{2n-2}^{n-1}/n)$，显然这个数量级的增长更为可怕。所以为了降低解码规模，从庞大的候选树中挑选出最有可能的结构树，我们可以将条件概率最大的那棵句法树做为句法分析的结果，即：

$$T_{best} = \underset{T}{\operatorname{argmax}} P(T|S,G) \qquad (2\text{-}1)$$

其中 T 树某句法树，S 是句子，G 是相应的文法。

为了找到条件概率最大的句法树，就要引入概率上下文无关文法

（Probabilistic Context Free Grammar，PCFG）的概念。基于概率上下文无关文法的句法分析方法，最早于 20 实际 80 年代被提出来，这种方法既有规则方法的特点，也运用了概率信息，可以认为是规则与统计方法的综合体。

概率上下文无关文法给上下文无关文法中的每一条推导规则 P 一个概率：

$$\mathrm{P}(\mathrm{A} \rightarrow \mathrm{B} \mid \mathrm{A}) \tag{2-2}$$

且利用最大似然估计该概率为：

$$\mathrm{P}(\mathrm{A} \rightarrow \mathrm{B} \mid \mathrm{A}) = \frac{\mathrm{Count}(\mathrm{A} \rightarrow \mathrm{B})}{\mathrm{Count}(\mathrm{A})} \tag{2-3}$$

其中 Count 表示统计其中的规则或标记的个数，则显然以 A 推导出所有的 B 的概率分布满足条件：

$$\sum_{B} P(A \rightarrow B|A) = 1 \tag{2-4}$$

表示 A 的所有推导可能的概率的和为 1。

在基于 PCFG 的句法分析模型中，其有三个假设条件：

（1） 位置不变性（place invariance）：子树的概率与该子树的儿子结点在子树中的位置无关。

（2） 上下文无关性（context-free）：子树的概率与子树外的结点无关。

（3） 祖先无关性（ancestor-free）：子树的概率与子树的祖先结点无关。

假设 T 是一个句法树，S 是输入的句子，则我们可以把句法树 T 的产生分解成一个自顶向下的过程：首先由一个初始的根结点产生它的子结点，非叶子结点的子结点继续产生子结点，反复如此，直至产生了所有的叶子结点。根据 PCFG 的三个假设条件，我们能很容易得到句法树 T 的产生概率就是将这些产生过程每一步的句法规则概率连乘起来，即

$$\mathrm{P}(\mathrm{T}, \mathrm{S}) = \prod_{r \in T} P(r) \tag{2-5}$$

举个例子，图 1-1 的句法树的 PCFG 的概率，可以表示成：

P(T,S) = P(IP->VP+PU|IP) P(VP->ADVP+VP| VP)* P(PU->。|PU)**
P(ADVP->AD|ADVP) P(VP->VV+NP|VP)* P(AD->全面|AD)* P(VV->推行|VV)* P(NP-> NP+NP|NP)* P(NP->NN+NN|NP)**
P(NP->NN+NN|NP) P(NN->教育|NN)* P(NN->收费|NN)**
P(NN->公示|NN) P(NN->制度|NN)*

根据常识，这种小概率的连乘很容易导致浮点数溢出，所以我们一般都将概

率转化成对数形式，由此句法树的概率公式（2-4）的对数形式如下：

$$\log(\mathrm{P}(\mathrm{T},\mathrm{S})) = \sum_{r\in T} \log(P(r)) \tag{2-6}$$

则求条件概率最大的那棵句法树的公式（2-1）可以转化为：

$$T_{best} = \underset{T}{\operatorname{argmax}}\, P(\mathrm{T}|\mathrm{S},\mathrm{G}) = \underset{T}{\operatorname{argmax}} \frac{P(T,S|G)}{P(S|G)} \tag{2-7}$$

对于某个具体的句子，其 P(S|G)表示由句法生成句子的概率可以认为是一个常数，则再根据公式（2-5）：

$$T_{best} = \underset{T}{\operatorname{argmax}}\, P(\mathrm{T},\mathrm{S}|\mathrm{G}) = \underset{T}{\operatorname{argmax}} \log(P(\mathrm{T},\mathrm{S}|\mathrm{G})) \tag{2-8}$$

我们可以将 P(T,S|G)理解成语言模型（language model），因为这个概率是定义在整个语言文法 G 上，且具有归一性质，即：

$$\sum_{T,S} \mathrm{P}(\mathrm{T},\mathrm{S}|\mathrm{G}) = 1 \tag{2-9}$$

2.1.3 PCFG 的三个基本问题

在运用基于概率的上下文无关文法时，我们会面对三个基本问题：

（1） 对于给定的句子 S 和上下文无关文法 G，快速计算句子的生成概率 P(S|G)；

（2） 对于给定的句子 S 和上下文无关文法 G，选择最佳的句法树 T，即求最大概率的结构树 T，$\underset{T}{\operatorname{argmax}}\, P(T|S,G)$；

（3） 对于给定的上下文无关文法 G 和句子 S，如果调节 G 的概率参数，使句子 S 具有最大的概率，即求$\underset{G}{\operatorname{argmax}}\, P(S|G)$；

第一个问题，在给定 PCFG 的情况下，我们可以利用内向算法（inside algorithm）和外向算法（outside algorithm）进行句子概率的快速计算，两个算法都是通过动态规划算法来推导出 S 所有可能的结构的概率最后累加起来，其中内向算法是自底向上的算法，外向算法是自顶向下的算法并利用到了前者。

第二个问题，上文提到，一个句子的所有可能的结构树的规模可以达到指数级别，显然我们不能对所有句法树一一进行概率计算，我们一般利用维特比算法（Viterbi algorithm）来对句子进行解码。维特比算法在句法分析中的具体实现即 CKY 算法，可以将指数规模的问题降至多项式复杂度。

第三个问题，针对 PCFG 参数估计问题，一般的思路是采用期望最大算法（Expectation-maximization）：给 G 的每个规则赋予一个随机的概率值，得到文法 G_0，然后根据 G_0 和训练树库，统计每个规则的使用次数的期望次数，用该期

望次数进行最大似然估计，得到语法 G 的新的参数估计 G_1。循环如此，直到 G 的概率参数

2.2 基于中心驱动模型的句法分析

2.2.1 中心驱动模型基本原理

由于在 PCFG 中没有考虑到上下文信息，导致基于 PCFG 的句法分析模型虽然简单、高效，但是对词汇信息不敏感从而使得其消歧能力较弱。

比如下图 2-1，两棵句法树描述的都是同一个句子“委员会由农业部组织组建”，这是两个应用相同句法规则，但结构不同的句法树，根据 PCFG，它们的概率应该是相等的。此时，则基于 PCFG 的句法分析无法判断哪棵树更优。

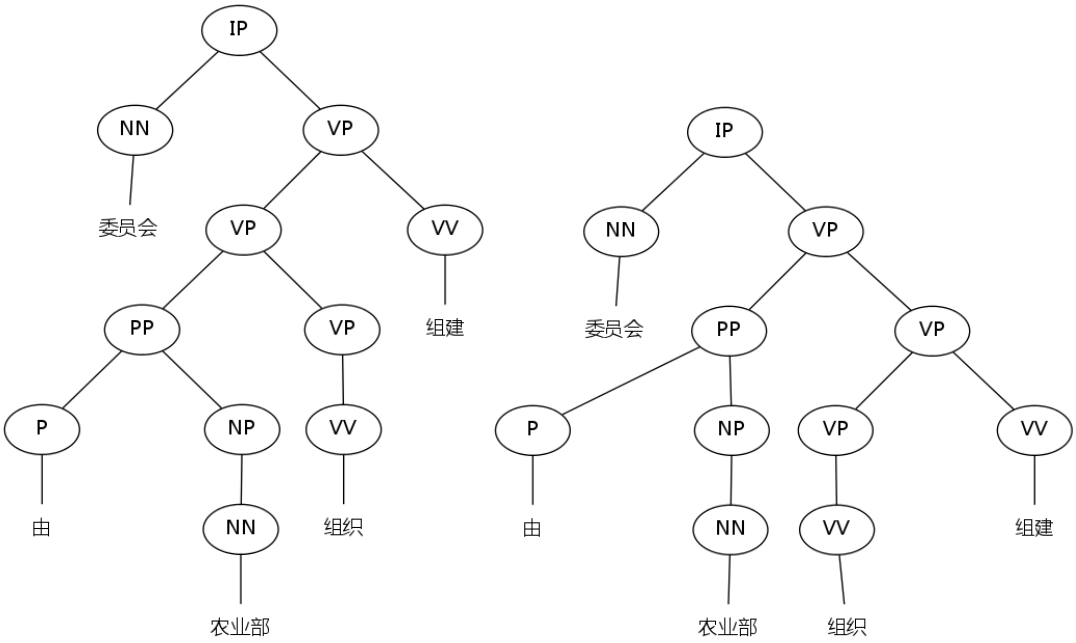

图 2-1 两棵具有相同句法规则，但是结构不同的句法树

针对 PCFG 这些问题，Collins 等人引入了基于 PCFG 的词汇化短语结构句法分析方法。这种方法的思想是：对句法树中的每个非终结符都挑选出一个中心词(或称核心词)，并用该中心词和其词性标记该非终结符，然后 CFG 中的每个规则的概率都要依据中心词进行估计。中心驱动模型是最具代表性的词汇化句法分析模型。

在中心驱动模型中计算句法树的概率的过程中，首先要确定句法树中每个成分的中心成分。标注中心成分的方法一般是根据一些语法规则，或者树库自带的中心词，本文会详细说明（见在本章 2.2.3）。

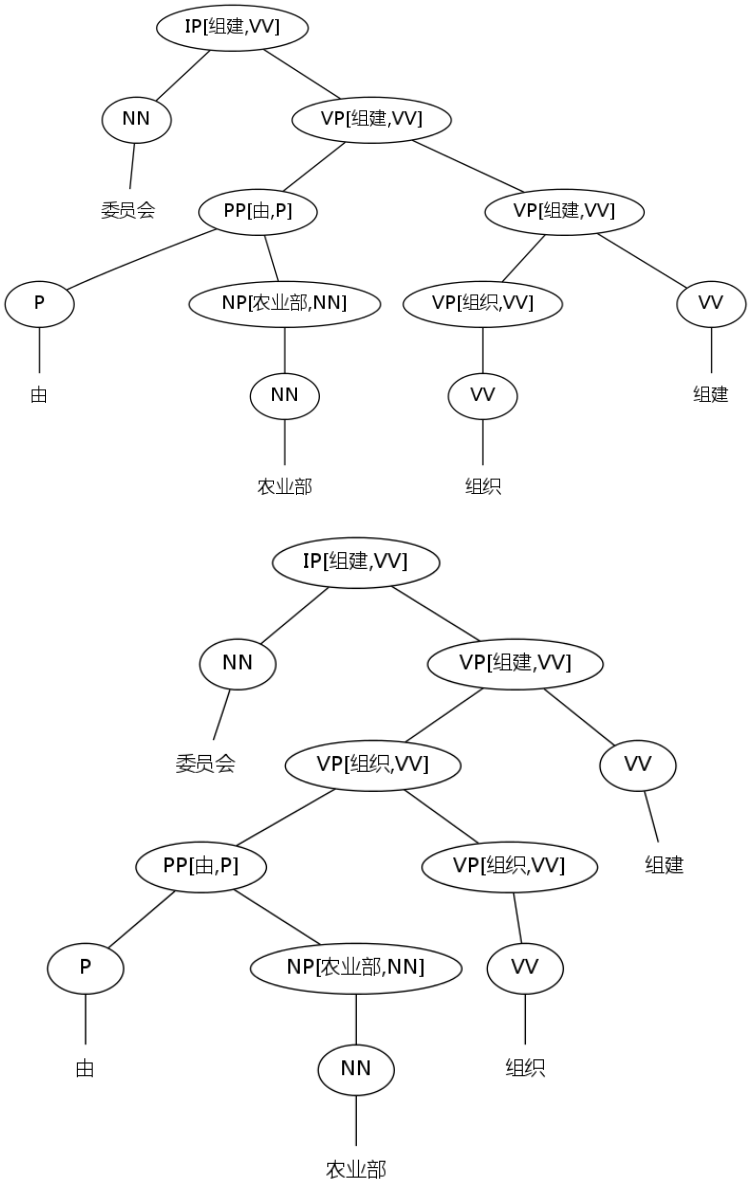

图 2-2 基于中心驱动的句法树表示

如图 2-2 所示的两棵句法结构树为图 2-1 的两棵句法树词汇化后的形式。下面举例图 2-2 中的句法树 a 中的一条规则 *VP->VV NP* 为例。在中心驱动模型下，该规则的词汇化形式如下：

VP(推行，VV)->VV(推行，VV) NP(制度，NN)

则利用最大似然估计，该规则的概率估计为：

$$P\left(VP(\textit{推行，}VV)\rightarrow VV(\textit{推行，}VV)NP(\textit{制度，}NN)\right)=$$

$$\frac{Count(VP(\textit{推行，}VV)->VV(\textit{推行，}VV)\ NP(\textit{制度，}NN))}{Count(VP(\textit{推行，}VV))}$$

此时再来观测图 2-2 中的两棵句法树的概率差异。去除它们共有的规则后，这两棵树的上下文无关概率的差异在于 $P(VP(\text{组建，}VV)\rightarrow PP(\text{由，}P)VP(\text{组建},VV))$和 $P(VP(\text{组织，}VV)\rightarrow PP(\text{由，}P)VP(\text{组织},VV))$，通过统计语料中这两个规则的概率我们就能判断这两棵树孰优孰劣。

由于引进词汇信息，所以利用该方法估计规则的概率将会面临非常严重的数据稀疏问题。Collins 针对这个问题引入了一系列独立性假设，他将每条词汇化的规则看作一个马尔科夫过程，将每个规则形成的过程分为三个阶段：

（1） 由父结点生成中心成分结点；

（2） 自右向左依次生成中心成分结点的左成分；

（3） 自左向右依次生成中心成分结点的右成分。

则，每条 CFG 规则可以改写成如下形式：

$$\begin{gathered}P(hw,ht)\rightarrow L_n(lw_n,lt_n)\ldots L_1(lw_1,lt_1)\\H(hw,ht)\\R(rw_1,rt_1)_1\ldots R_m(lw_m,lt_m)\end{gathered}\qquad(2\text{-}10)$$

上式中，P 为 CFG 规则的左端非终结符，H 为 P 结点的中心成分，L_i 为中心成分的左修饰成分，R_i 为右修饰成分，hw、lw、rw 皆为相应成分的中心成分，ht、lt、rw 为他们的词性。如下图所示：

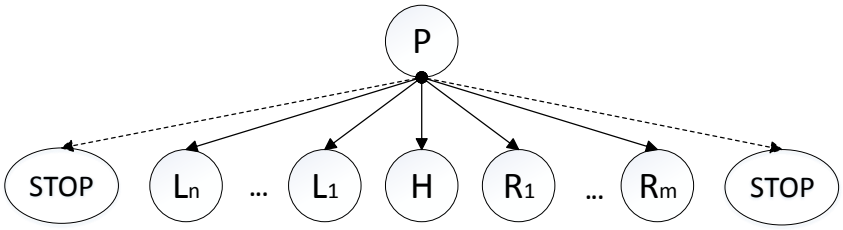

图 2-3 Collins 的独立性假设示意图

上图 STOP 为左右两边的停止符号，根据 Collins 的三条假设：

（1）由父结点 P 和中心词生成中心成分 *H(hw,ht)*的概率为：

$$P_h(H(hw,ht)|P,h_w,h_t)\qquad(2\text{-}11)$$

（2）由父结点和中心词成分和中心词，依次生成中心成分左边的非终结符

的概率和右边的非终结符概率分别为：

$$\prod_{i=1}^{n+1} P_l(L_i(lw_i, lt_i)|P, h_w, h_t, H) \tag{2-12}$$

$$\prod_{i=1}^{m+1} P_r(R_i(rw_i, rt_i)|P, h_w, h_t, H) \tag{2-13}$$

综上，公式 2-10 规则的概率为：

$$P = \prod_{i=1}^{n+1} P_l(L_i(lw_i, lt_i)|P, h_w, h_t, H) * P_h(H(hw, ht)|P, h_w, h_t) * \prod_{i=1}^{m+1} P_r(R_i(rw_i, rt_i)|P, h_w, h_t, H) \tag{2-14}$$

其中，L_{n+1} 和 R_{m+1} 为结点左右两边的停止符号。

2.2.2 中心驱动模型概率参数估计

使用独立性假设后，词汇化规则可以分解成较小的三个部分进行概率计算，但是针对分解后的三个部分进行最大似然估计时仍然会面临数据稀疏问题，因此需要引入语言模型中平滑的方法对每个部分的概率进行估计。Collins 通过三级回退模型利用插值的方式来进行平滑处理：

$$P = \lambda_1 e_1 + (1 - \lambda_1)(\lambda_2 e_2 + (1 - \lambda_2) e_3) \tag{2-15}$$

其中 e_1，e_2 和 e_3 分别表示三种级别的回退模型，λ_1 和 λ_2 分别为 e_1 和 e_2 的权重，每级的回退模型的计算方式如下表，P_r 和 P_l 同：

表 2-1 三级回退模型

回退模型	$P_h(H(hw, ht)\| \ldots)$	$P_l(L_i(lw_i, lt_i)\| \ldots)$
e_1	P, h_w, h_t	P, h_w, h_t, H
e_2	P, h_t	P, h_t, H
e_3	P	P, H

对于公式 2-15 中的 λ_1 和 λ_2 不是随意设置的，λ_1 是要通过计算得来，对于计算 P_h 时，P_{h1} 和 λ_1 的计算公式如下：

$$P_{h1} = P_h(H(hw, ht)|P, h_w, h_t) = \frac{Count(H, P, hw, ht)}{Count(P, hw, ht)} \tag{2-16}$$

$$\lambda_1 = \frac{Count(P, hw, ht)}{Count(P, hw, ht) + \alpha U(P, hw, ht)} \quad (2\text{-}17)$$

其中 U 为定义在 *P*，*hw*，*ht* 确定下，有多少不同的 H，即中心成分的数量，α 为超参数。P_{h1} 和 λ_1 的相乘的值对 H 求和不为 1，且损失的概率为：

$$1 - \sum_H \lambda_1 * P_{h1} = 1 - \frac{Count(P, hw, ht)}{Count(P, hw, ht) + \alpha U(P, hw, ht)} \quad (2\text{-}18)$$

$$= 1 - \lambda_1$$

结合公式 2-15 可以看出该损失概率为二级概率的权值，从而保证了回退模型的概率的归一性。类似地，对于二级概率也是同样的方法得到三级概率的权重为$1 - \lambda_2$，由于第三级为最后一级，则不需要计算λ_3。但是对于超参数 α，则需要利用开发集确定。

2.2.3 中心成分的确定

中心驱动模型重要的一步，就是确定每个句法成分的中心成分。如果树库的标记没有标注中心成分，则我们可以根据一系列中心词决策表来确定中心成分。其中 Xia[38]在 2001 年提供的英文和汉语的中心成分决策表是被广泛使用。Xia 的决策表主要是根据 CTB 的标注体系来进行的，所以其他标注集的树库可能需要进行适当转化标记。其决策表中的每个条目都是有一个三元组构成，形式如下：

<x，direct，$y_1/y_2/.../y_n$>

其中 *x* 表示父结点标记，可以是词性或者句法标记；*direct* 值可为 right 或者 left，表示向左开始或向右开始寻找中心成分；$y_1/y_2/.../y_n$ 是中心成分优先列表。则该三元组的意思为：对于标记为 *x* 的父结点，它的中心词成分为按照 *direct* 方向寻找到的第一个出现在 $y_1/y_2/.../y_n$（不分先后）中的标记的子结点的中心成分。由此可得确定一棵句法树的所有句法成分的中心成分的过程是一个递归的过程，且需要首先确定叶子结点的中心成分。一般的叶子结点的中心成分为其自身。

Xia 的汉语中心成分决策表中共有 21 个条目，附录 4 是其中汉语中心成分表中的所有条目。

举例说明，表中条目<*NP, right, NP/NN/NT/NR/QP*>表示：对于一个名词短语 NP，它的中心成分为从其子结点最右端开始找，第一个标记在“*NP/NN/NT/NR/QP*”中的子结点的中心成分，如图 2-2 句法树 a 中管辖“*农业部和卫生部*”的 NP 结点，其中心成分为管辖“*和卫生部的*”的中心成分，即“(*NN 卫生部*)”。<*PP, left, PP/P*>表示对于一个介词短语 PP，其中心成分为其子结点最左端开始找，第一个子结点标记为 *PP* 或 *P* 的结点的中心成分，如图 2-2 句法树 a 中的 PP 结点的

中心成分即为“*(P 由)*”。又如，<*VV, right, VV*>则表示 VV 词性的标记其中心成分为其自身。

2.3 其他句法分析模型

2.3.1 层次化汉语长句结构分析模型

对于完全句法分析算法，一般的线图分析算法和 Earley 等算法虽然将句法分析的复杂度大大缩减到了 $O(N^3)$（N 为句子的词数），但是随着句子长度增加到 20、30 甚至 50，三次方的复杂度依然大大制约着句法分析的效率。李幸[39][40]等人发现，长句分析的困难不单是体现在算法的复杂度上，早期的子句句法分析错误往往会引起整个长句分析的崩坏，从而影响句法分析的准确率和召回率。因此李幸等人根据汉语标点符号的使用，提出层次化句法分析方法（hierarchical parsing, HP）。

HP 算法主要分为三个步骤：

（1） 对包含可“分割”的标点符号的长句进行分割；

（2） 对分割后的每个子句进行句法分析，得到子树（子句内部分析）；

（3） 对各个分析后的最大概率的子树进行句法分析（子句间的分析），得到最后的完整的句法分析树。

整个算法的示意图如下图 2-4 所示：

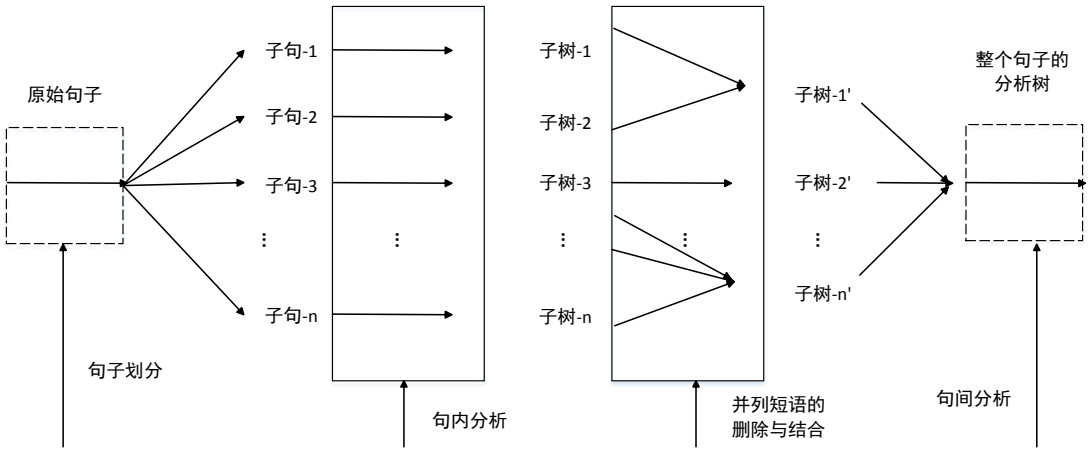

图 2-4 HP 算法示意图

对于第一步，HP 方法中定义的“分割”标点须要满足以下的条件：该标点分隔开的子句单元是以整体为单位与其他子句单元发生关系的，即，子句内部的部分成分不与其他子句发生关系，那么这种标点就被定义为“分割”标点，其余的标点为“普通”标点。

HP 算法主要将逗号、冒号和分号作为“分割”标点。值得注意的是，不是所有的逗号都可以直接作为“分割”标点。根据国家标准《标点符号用法》[41]，逗号的主要用于连接主语与谓语、动词与宾语、状语和其修饰的句子以及复句内的各个分句。其中只有最后一种情况下，逗号可以作为句子分割，而其他三种情况需要具体考虑逗号左右的两个成分是否相对独立。

对于第二步，子句内部的分析，HP 算法采用图表分析算法。在该阶段分析中，句法分析的初始输入是各个子句所包含的词和词性标注，然后利用句法分析算法可以得到概率最大的子树。接着还要判断逗号左右的子树是否为并列关系，如果是则合并。最后利用这些子树进行下一步分析。

对于第三步，子句间的分析和子句内的分析算法一致，只是前者的输入可能为各个子句生成的结构树的结构标记和分割的它们的标点符号。经过第三步，得到的最后的结果就是概率最大完整的句法分析树。

HP 利用“分而治之”的策略大大减少了时间消耗，并改善了句法分析对长句的句法分析效率。

2.3.2 基于移进归约的确定性依存分析模型

在自然语言处理中，我们有时只需要了解句子中的每个词之间的依存关系，而无需了解句子内部的结构关系。这种根据词之间的依存关系描述的语言结构的框架叫做依存语法（dependence grammar）。Gaifman[42]在 1965 年证明了依存语法与上下文无关文法时等价的。

在依存语法理论中，“依存”指词之间支配与被支配的关系。依存关系是有方向的，箭头的指向就是被支配的词。在实际引用上还会为了丰富依存结构的句法信息，在箭头上添加不同的句法标记。

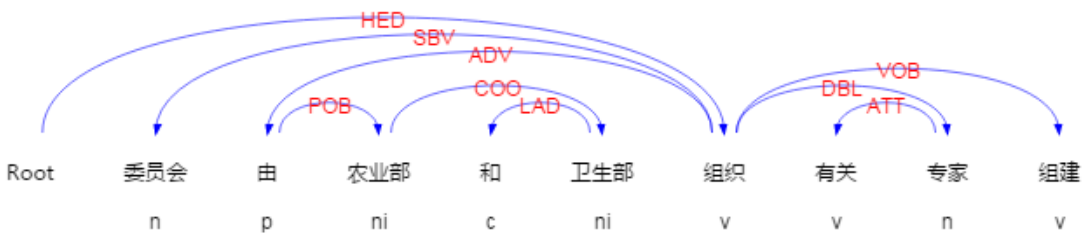

图 2-5 依存句法结构图式

确定性依存分析方法是目前比较主流的依存句法分析方法，该方法以特定的方向逐个取出待分析的词，然后为每个词产生一个单一的结果，直到处理完最后一个词。

Yamada[43]在 2003 年提出基于移近归约（Shift-Reduce）算法的确定性依存分析方法。该方法对待分析的词提供共三个不同的动作：Shift（移近）、Right-Reduce（右归约）、Left-Reduce（左归约）。在实现该算法的时候，会利用一个二元状态<S,Q>表示目前的依存状态，其中 S 是堆栈，用于储存结点，Q 则是用于储存未处理的词序列，如下图 2-6 所示：

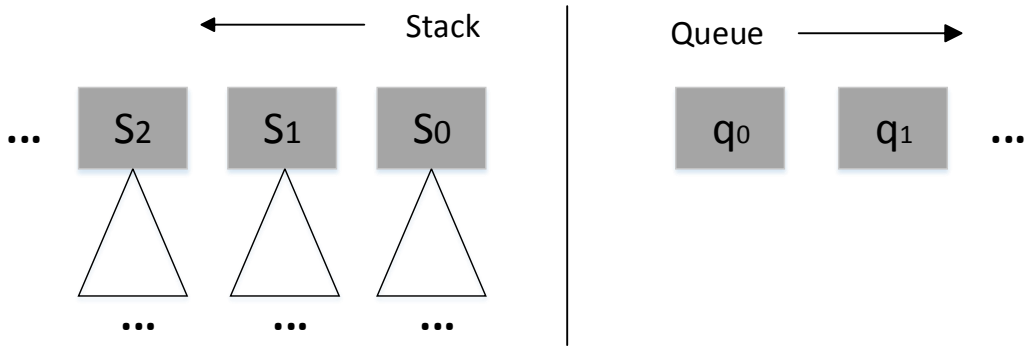

图 2-6 移进归约方法的状态示意图

则结点的三个动作的具体操作如下：

（1） Shift：移动队列 Q 中的第一个词 q_0 到堆栈 S 的中，生成新的只有单一词的结点 s_0。

（2） Right-Reduce：将堆栈 S 中的 s_0 和 s_1 结合起来，且 s_0 支配 s_1，并生成新的结点 s_0。

（3） Left-Reduce：将堆栈 S 中的 s_0 和 s_1 结合起来，且 s_1 支配 s_0，并生成新的结点 s_0。

该算法的分析过程是从左到右，每次通过 Shift 取右边第一个新词，然后再在窗口内（一般取堆栈 S 前两三个结点和队列 Q 的前两个词）提取特征，如下表 2-2。接着通过机器学习利用这些特征分析下一步的动作，如果还是 Shift，则继续提取一个新词，并移动窗口，如果是 Reduce 则将需要归约的相邻的两个结点合并成一个结点。所以最后在堆栈 S 内会只剩下一个结点，即根结点，而队列 Q 则为空时依存分析完毕。

表 2-2 Yamada 的移近归约方法的特征

类型	值
pos	当前词的词性
lex	当前词本身
ch-L-pos	当前词的左支配的结点的词的词性
ch-L-lex	当前词的左支配的结点的词的词
ch-R-pos	当前词的右支配的结点的词的词性
ch-R-lex	当前词的右支配的结点的词的词

这种确定性的依存句法分析方法在进行训练和具体决策的时候都只是局部最优。该方法用局部最优来近似全局最优的方法会导致错误传递，使其在准确率上弱于其他生成式和判别式的方法。但是其的优点也是不容忽视的，线性的时间复杂度和丰富的特征利用是其最大的优点，这些优点值得其他句法分析方法学习。

2.4 条件随机场模型

在自然语言处理任务中，我们经常使用条件随机场模型（Conditional Random Fields，CRF）进行序列标注任务。条件随机场是一种在给定随机变量 X 条件下，计算输出的变量 Y 的马尔科夫随机场或无向图模型，由 Lafferty[44]等人于 2001 年提出。目前，条件随机场被广泛应用在自然语言处理任务，比如分词、词性标注、新词发现和命名体识别等任务，并且效果良好。本文实现的句法分析系统利用了条件随机场进行了句子分块、句法结构分析和句法标记识别的任务。

2.4.1 条件随机场的定义

若 G=(V,E)为无向图，其中 V 为结点集合，E 为无向边集合。$Y = \{Y_v | v \in V\}$，即 V 中的每个结点都会对应一个随机变量Y_v，Y_v的取值范围是{y}。如果在已知随机变量 X 的观测下，每个随机变量Y_v若都满足以下马尔科夫特性：

$$p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v) \quad (2\text{-}19)$$

其中w~v表示 w 与 v 为两个相邻的节结，则 p（Y|X）为一个条件随机场。

特别地，如果每个随机变量Y_v是以顺序的方式在图 G 中链接而成，如图 2-7，则称该图模型是线性链条件随机场，线性链条件随机场是最常见和简单的条件随机场模型，主要用于解决序列标注问题。

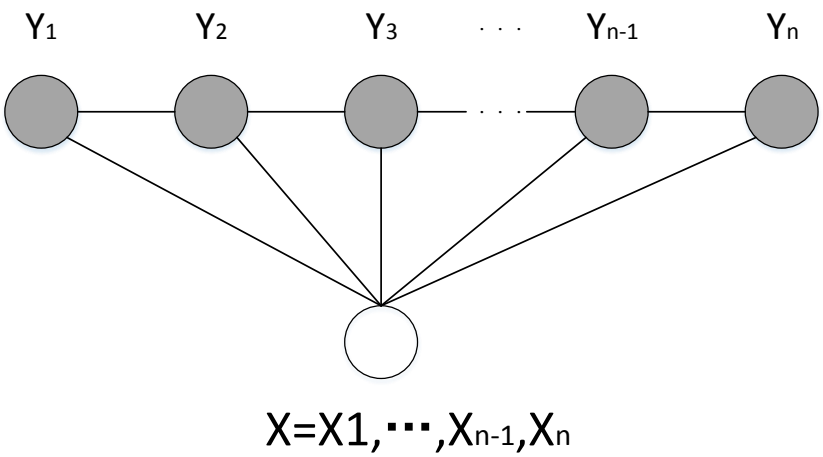

图 2-7 线性链条件随机场结构图

在线性条件随机场模型中，随机变量 X 作为观测序列，不同 X 之间没有直线连接即不存在图结构。

2.4.2 条件随机场的参数估计

若 P(Y|X)是线性条件随机场，在给定观测序列 X 为 x 时，对于随机变量 Y 的条件概率可以表示为：

$$\mathrm{p(y|x)} = \frac{1}{Z(x)}\exp(\sum_{i,k}\lambda_k\, t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l\, s_l(y_i, x, i)) \tag{2-20}$$

其中 Z(x)为归一化因子，表示所有可能的序列输出的概率求和，其值为：

$$\mathrm{Z(x)} = \sum_{y}\exp(\sum_{i,k}\lambda_k\, t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l\, s_l(y_i, x, i)) \tag{2-21}$$

上式中，t_k和s_l是特征函数，λ_k和u_l是对应的权值。t_k是转移函数，表示在观测值 X 条件下，标注序列 Y 在 i 和 i-1 的位置上的标记的转移概率。s_l是状态函数，表示观测值 X 在位置 i 上的标记概率。λ_k和u_l的权值需要在训练条件随机场模型的过程中不断修正而估计出来的，是模型的主要参数。

通常下，转移函数和状态函数的取值为 0 或者 1，当位置 i 和 i-1 的观测变量 x 和状态y_{i-1}和y_i满足条件时，它们的值为 1，否则为 0。所以可以统一转移特征和状态特征为：

$$f_k(y_{i-1}, y_i, x, i) = \begin{cases} t_k(y_{i-1}, y_i, x, i), & k = 1,2,\dots K_1 \\ s_l(y_i, x, i), k = K_1 + l; l = 1,2,\dots, K_2 \end{cases} \tag{2-22}$$

K_1和K_2表示特征函数的个数，然后对统一的新特征函数在各个位置求和，记为：

$$f_k(y,x)=\sum_{i=1}^{N} f_k(y_{i-1},y_i,x,i),k=1,2,\dots K \tag{2-23}$$

此时，K表示所有的特征函数的个数，则可以用w_k表示特征$f_k(y,x)$的权值，则式 2-20 的条件随机场条件概率可以改写为：

$$\mathrm{p(y|x)}=\frac{1}{Z(x)}\exp\sum_{k=1}^{K} w_k f_k(y,x) \tag{2-24}$$

$$\mathrm{Z(x)}=\sum_{y}\exp\sum_{k=1}^{K} w_k f_k(y,x) \tag{2-25}$$

用 w 表示权值的向量，即：

$$\mathrm{w}=(w_1,w_2,\dots,w_K)^T \tag{2-26}$$

用 F(y,x)表示全局特征向量，即：

$$\mathrm{F(y,x)}=(f_1(y,x),f_2(y,x),\dots,f_K(y,x))^T \tag{2-27}$$

此时，条件随机场还可以写成如下的形式：

$$p_w(\mathrm{y|x})=\frac{1}{Z(x)}\exp(w\cdot F(y,x)) \tag{2-28}$$

$$Z_w(\mathrm{x})=\sum_{y}\exp(w\cdot F(y,x) \tag{2-29}$$

在给定数据集后，条件随机场可以通过极大似然或正则化的极大似然估计进行模型的学习，主流的优化算法有改进的迭代尺度 IIS、梯度下降和拟牛顿法的 BFGS 算法。本节不会对具体的算法的证明和实现做过多的描述，有意向的读者可以参考李航博士的《统计学习方法》[45]。

改进的迭代尺度法是通过极大化训练数据的对数似然函数直至收敛，最终得到模型的参数。条件随机场模型也可以通过牛顿法和拟牛顿法进行参数的优化。拟牛顿法可以大大减少优化过程的计算量，其中目前最常用的是有限内存下的 BFGS 方法（limited memory BFGS，LBFGS）。该方法通过有限量的储存空间来近似存储的二阶梯度信息，大大提高了条件随机场的训练效率。

对于序列标注问题，条件随机场模型作用是：在给定观测序列 x 情况下，预测最大的条件概率的输出序列 y^*。由于归一化因子 Z(x)与 Y 的输出无关，所以我们可以将条件随机场的概率公式简化成只需计算非规范化概率。条件随机场的预测过程的解码算法使用的是维特比算法。

2.4.3 条件随机场工具包

条件随机场有很多实现工具，比如 CRF++、Pocket CRF、CRFSuite 等。下面来简要介绍了这三个比较主流的工具包的特点：

（1）CRF++[46]。CRF++可以认为是目前使用最广泛的条件随机场工具，许多其他的工具也都是从该工具包发展而来的。该工具包十分灵活，可以自行制定系统需要的特征模板、设置正则算法和其他超参数，支持 window 和 linux 双重环境。本文采用了 CRF++来进行序列标注的相关工作。

（2）Pocket CRF[47]。Pocket CRF 是 CRF++的一个修正版，它在包含了 CRF++所有的基本特性的基础上，还可以处理高阶特征。它的操作方式和特征模板的设置和 CRF++基本一致，只是在特征模板上需要在每个特征模板的结尾加上%y[0]，这种设置其实更符合 CRF 的特征函数的形式。

（3）CRFSuite[48]。CRFSuite 是一个由纯 C 语言实现的条件随机场工具，所以在训练速度上较 CRF++快，其他方面如特征模板、正则算法和 CRF++类似。值得注意的是 CRFSuite 有 Python 的 API，并已经提供到了 sklearn-crfsuite 上。所以 CRFSuite 比较适合纯 Python 语言开发的自然语言处理任务。

2.5 本章小结

本章主要描述本文实现的句法分析系统中需要的基础知识，主要包括概率上下文无关文法、基于中心驱动模型的句法分析分析、层次化汉语长句结构分析、基于移近归约的确定性依存句法分析以及条件随机场模型。本文实现的句法分析系统结合了上述的三个句法分析模型的特性，并利用条件随机场进行具体任务的实现。

第三章 基于句子分块的汉语句法分析

3.1 引言

近几年，汉语句法结构分析得到了很大的发展，但是仍然存在一些不可忽视的问题。一个重要的问题是：在对长句进行分析时，基于中心驱动模型的句法分析效率会大大降低。为此，本文借鉴层次化汉语长句结构分析的特点，将长句根据一些策略进行分割，然后对每个子句分别进行结构分析生成子树，最后合并这些子树，得到完整的句法分析树。在结构分析的过程中，本文进一步在句法分析模型中结合基于移近归约依存分析模型丰富特征的特性，增加了结构分析的准确率。最后利用集束搜索技术进一步改善句法分析的效果。

本章介绍了基于句子分块的汉语句法分析系统，主要包括数据的预处理、句子的分块、基于条件随机场的句法结构分析和集束搜索，并就各种特征和不同的设置展开实验。

3.2 基于句子分块的汉语句法分析

基于句子分块的汉语句法分析的整体框架如下：

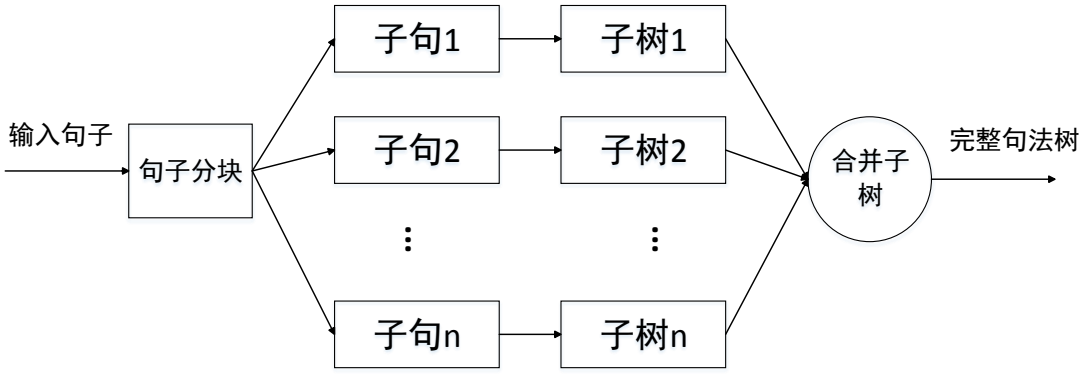

图 3-1 基于句子分块的汉语句法分析模型框图

其中输入句子的格式是已切分和词性标注的句子，其中切分和词性标注的标

准皆为 CTB 的标准。

下面将在每个小节介绍框架中每个小模块的具体处理方法，并在 3.3 章节介绍句法分析模型的实验。

3.2.1 预处理

预处理模块主要应用于模型训练前，主要包括去除冗余标签、中心词标注和二叉树的转化。

去除冗余标签。本文使用的语料是 CTB8.0，在 CTB 中的存在一些特殊标记如 SBJ、NONE 之类的语法标签。目前本文的句法分析系统没有使用这些标签，所以需要去除这些标签。

首先需要删除空结点，即一些不包含任何词语的结点，如图 3-2 的句法树中的 WHNP-2 和 NP-SBJ 结点。图 3-3 为删除空结点后的句法结构图。

```
(NP (NP (NN 早期))
    (CP (WHNP-2 (-NONE- *OP*))
        (CP (IP (NP-SBJ (-NONE- *T*-2))
                (VP (VV 缺乏)
                    (NP-OBJ (NN 系统性))))
            (DEC 的))))
    (QP (CD 单)
        (CLP (M 个)))
    (NP (NN 投资)))
```

图 3-2 未删除空结点的句法树

```
(NP (NP (NN 早期))
    (CP (CP (IP (VP (VV 缺乏)
                    (NP-OBJ (NN 系统性))))
            (DEC 的))))
    (QP (CD 单)
        (CLP (M 个)))
    (NP (NN 投资)))
```

图 3-3 删除空结点后的句法树

接着，还要删除一些不需要的文法、语义标签，如“-SBJ”、“-OBJ”等，图 3-2 中的结点“NP-SBJ”处理后转换为“NP”，得到上图 3-4。

```
(NP (NP (NN 早期))
    (CP (CP (IP (VP (VV 缺乏)
                    (NP (NN 系统性))))
            (DEC 的))))
    (QP (CD 单)
        (CLP (M 个)))
    (NP (NN 投资)))
```

图 3-4 删除文法、语义标签后的句法树

最后删除了上述结点，有可能会产生一些 X->X 的冗余结点，比如上图 3-4 中的 CP->CP，这些无意义的分支，需要删除，得到如下图 3-4：

```
(NP (NP (NN 早期))
    (CP (IP (VP (VV 缺乏)
                    (NP (NN 系统性))))
            (DEC 的))))
    (QP (CD 单)
        (CLP (M 个)))
    (NP (NN 投资)))
```

图 3-5 删除冗余结点后的句法树

中心成分的标注。为了增加句法分析的表达能力和特征的丰富度，本文的句法分析系统也进行了中心成分的标注，中心成分标注的方法主要参考(Xia,2001)的论文中的汉语中心成分决策表（具体详见 2.2.3）。经过中心成分的标注，图 3-5 的句法树转化成如下图 3-6 所示：

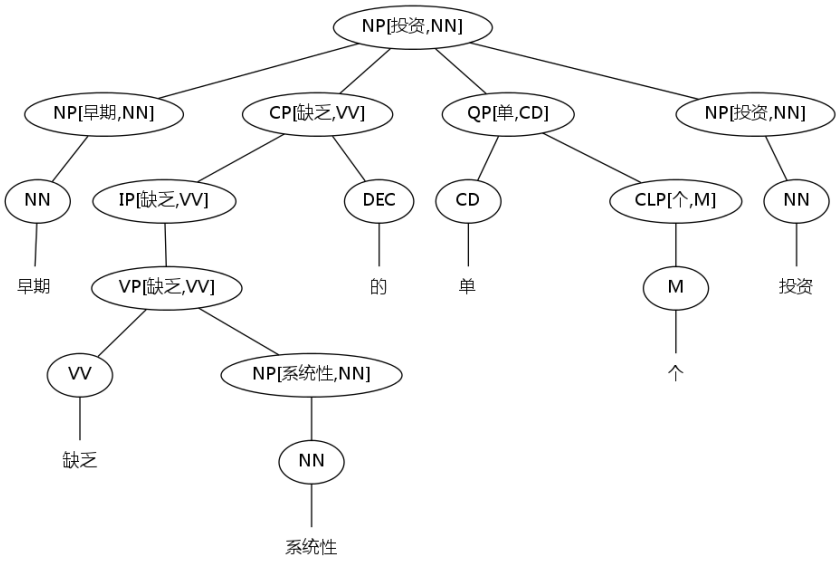

图 3-6 带中心成分标注的句法树

二叉树的转化。CTB 语料的句法树是多叉树结构。因为二叉树可以极大的降低句法分析复杂度，所以本文将 CTB 中的句法树全部转化成二叉树结构树。注意本文不会对单叉树进行处理，即不会将其转化成二叉树形式。下面简要介绍二叉树转化的方法。

根据中心成分。根据乔姆斯基的 X 阶标理论（X-bar theory）[49]，阶标可以分为若干个层次，处于最低层次的词 X 就是中心成分，中心成分左右带有若干个补足语，中心成分管辖着这些补足语。因此在转化成二叉树的过程中，我们一般将中心成分放在最底层，且先从左开始二叉化，当左边的依附的成分完成二叉化后，再进行右边成分的二叉化。在这期间生成的新的结点的标记为原先的父结点的标记加上星号，用以区分原来的父结点标记，显然这些新结点的中心成分和父结点的一致。如图 3-7 为图 3-6 的句法树的二叉化后的形式。

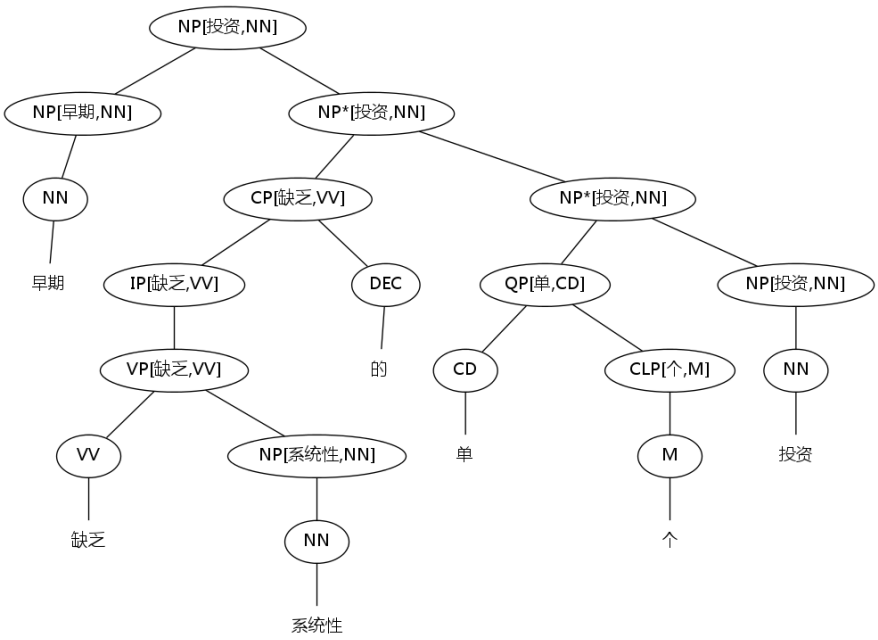

图 3-7 二叉化后的句法树

特殊处理。在遇到一些特殊句式时，二叉化需要特殊处理，否则很容易引起一些句法错误。比如对于“*《满城尽带黄金甲》*”，这类书名号或者引号包含起来的句式，本文需要先将中间的成分进行二叉化，然后再包含左符号，最后是右符号。再如，对于连词连接的两个成分，需要先各自二叉化后再一起处理，避免出现割裂的情况。但是在连词和顿号连用的句式，需要将连词和顿号等价处理，否则会出现错误。比如下图 3-8 是“*(NP (NN 集资) (PU 、) (NN 法制) (NN 建设) (CC 与) (NN 开发))*”的二叉化处理：

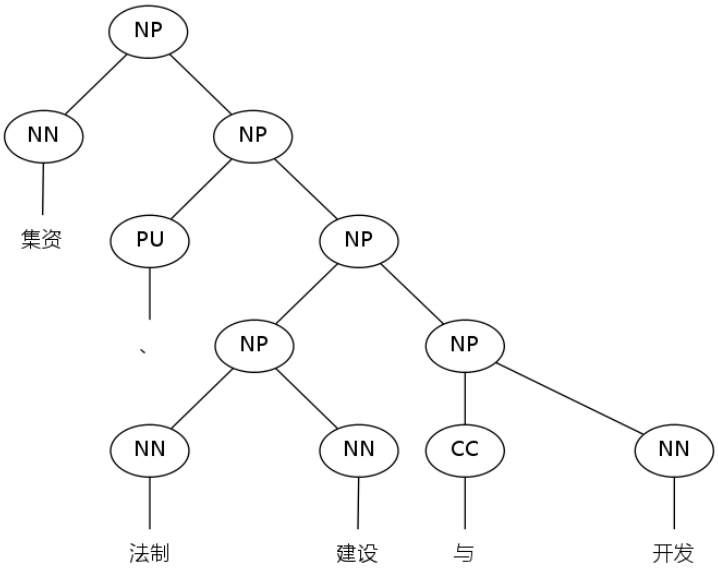

图 3-8 连词和顿号并现时的二叉化处理

3.2.2 句子分块

汉语是一种表意型的语言，所以汉语为了表达一个完整的意思，可能会通过逗号或者其他符号或者连接词将一连串的简单句连接成一个复杂的长句。这些长句一般词的数量都大大超过 20 个，这种长度的句子无疑大大加重了句法分析模型的负担。基本上所有的句法分析模型对于长句，其分析的效率和性能都大大降低了。但是反观这些复杂句中的简单句，它们其实每个句子间很大部分是完全独立的，特别对于一些流水句，只是罗列一些简单事件。为了解决以上问题，很多学者提出了层次化的汉语长句结构分析的方法，利用包含标点符号的文法规则对长句进行分割。本文借鉴了上述的方法，开发了一个基于序列标注的汉语的句子分块模块。

本文分块的思想借鉴了 G. Nunberg[50]的思想，本文的分块的具体思想是：如果一个由一定的数量的成分组成的短语或短句，满足其内部的词之间的关联紧密，而与周边的其他成分中的词关联较弱，则可以将这几个成分组成一个块。所以通过中心成分的划分，不同的块之间可以由中心词联系起来，从而组成一个完整的句子。而分块的作用是可以将长句分而治之，降低句法分析的规模，起了类似浅层句法分析的作用。

本文将分块问题转化成序列标注问题，利用条件随机场进行序列标注任务的实现。汉语中的句号、问号和感叹号是一个完整的句子的结束的标记，显然，我们可以将这些符号作为句子的分隔符。其他的符号如逗号、冒号也经常作为长句中的简单句的连接标记，但是不是所有的逗号和冒号都是可以作为简单句的分割

符的，需要考虑符号左右两侧的成分是否独立，如下列的两个句子：

句子 a：*内阁府/九日/公布/的/外交/舆论/调查/，/于/十月/五日/至/十五日/针对/全国/三千/名/二十/岁/以上/成年/男女/进行/调查/，/其中/一千七百零四/人/作出/回应/，/回应率/为/百分之五十六点八/。*

句子 b：*马英九/表示/，/在/过去/的/八/年/中/，/每次/参加/圣诞节/报/佳音/活动/时/，/无论/走在/万华区/、/天母/、/士林/夜市/以及/信义/商圈/或者/其他/市区/的/街道/上/，/都/深深/感受/市民/朋友/对/市/政府/的/热情/支持/与/期待/。*

句子 a 中每个逗号分隔开的短句都是独立的，即可以直接利用逗号分开。句子 b 中，“*表示*”管辖的成分是之后的所有词，所以如果直接的按照逗号分块则会导致“*马英九*”与“*表示*”成为一个块，所以需要做些特殊处理。这两个句子的分块结果如下（块的分界符是“##”）：

句子 a 分块结果：*内阁府/九日/公布/的/外交/舆论/调查##，##于/十月/五日/至/十五日/针对/全国/三千/名/二十/岁/以上/成年/男女/进行/调查##，##其中/一千七百零四/人/作出/回应##，##回应率/为/百分之五十六点八##。*

句子 b 分块结果：*马英九##表示##，##在/过去/的/八/年/中##，##每次/参加/圣诞节/报/佳音/活动/时##，##无论/走在/万华区/、/天母/、/士林/夜市/以及/信义/商圈/或者/其他/市区/的/街道/上##，##都/深深/感受/市民/朋友/对/市/政府/的/热情/支持/与/期待##。*

类似“*表示*”之类的词还有“*指出*”、“*预计*”、“*说*”、“*称*”、“*发现*”等，这些词经常出现在逗号前，导致在划分块的时候容易出错。这些问题不是可以简单地通过规则来解决，需要一种自动化的方法发现这些问题，并学习规则，而条件随机场可以学习到序列标注的前后规则，符合这些要求。

本文提出利用序列标注的方法来学习这些规则。本文的思路是：先利用训练集的句法树将句子按分割标点符号，主要即逗号和分号，将句法树拆分成子树，确保子树中没有这些分割符号。接着提取出这些子树的词和词性并打上序列标注标签，利用条件随机场学习这些数据。最后在测试时输入切分和词性标注后的句子，模型即可得到相应的子块。具体的拆分子树的算法是：

输入：CTB 训练集中的一棵完整的句法树；

输出：只带有词和词性的分完块的句子。

步骤：

步 1 将整棵树的根结点存在一个列表 L 中，设列表 R 用于存分完块的子树的根结点；

步 2 若 L 为空，则转步 4，否则从 L 的头部中取出一个结点 n，执行以下操作：

① 若结点 n 不包含分割符号或者结点是一个只有分隔符号构成的结点，则将结点 n 存入 R，返回步 2；

② 否则，将结点 n 拆分成 n_1 和 n_2（假如是二叉，单叉则只有 n_1），并存入 L 的头部，并转步 2；

步 3 将 R 中的所有的结点按顺序转成词和词性的格式输出，结束。

本文利用分完块的 CTB 训练集，只利用其的词和词性信息，训练一个条件随机场模型。模型的设置如下：

模型的标注标签采用的是“B”、“B_2”、“M”、“E_2”、“E”、“S”这六个标签，如表 3-1 所示。

表 3-1 分块的 CRF 模型的标签标注说明

块长度	标注序列
1	S
2	BE
3	BB_2E
4	BB_2E_2E
5	BB_2ME_2E
≥6	$BB_2M…ME_2E$

模型的特征只有词和词性，标记词特征为 WF，词性特征为 PF。词特征模板为 WF_{-2}、WF_{-1}、WF_0、WF_1、WF_2、$WF_{-2}WF_0$、$WF_{-1}WF_0$、WF_0WF_1、WF_0WF_2、$WF_{-1}WF_1$、$WF_{-1}WF_0WF_1$。词性特征模板和此特征模板一致，在此不再赘述。

3.2.3 句法分析

本文借鉴移近归约依存句法分析的方法中将句法分析转换成分类的思路，进一步将句法分析转换成序列标注问题，利用条件随机场的长距离依赖性和交叠性的特性并结合丰富的特征用以学习句法规则。

首先，首当其冲的问题是如何设置序列的标签体系，如果直接使用短语标签作为序列标签，而不同短语标签共 22 种，则至少需要近 44 种序列标签（如果只是用 B 和 O 两种标记）进行条件随机场的训练，这为条件随机场模型带来很大的负担。故本文采用两个子模块来进行句法分析的任务。一个是结构分析模块，即负责判断两个结点或一个结点是否生产新的结点；另一个是标签分类模块，即为新的结点打上句法标签。具体的流程图如下：

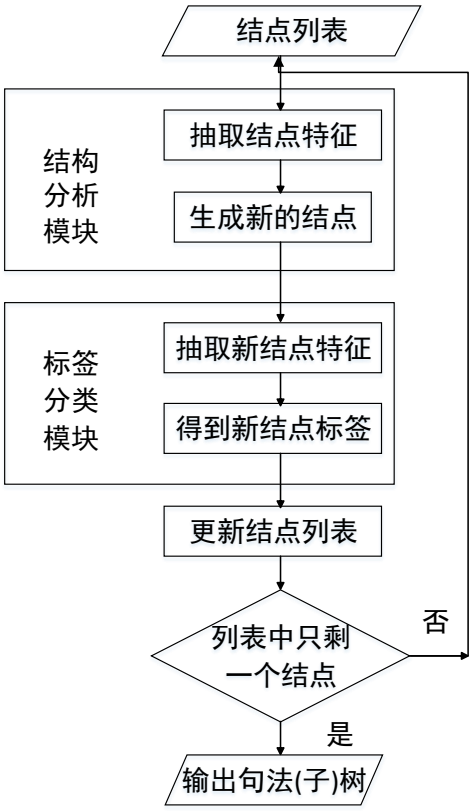

图 3-9 句法分析的流程图

这里，由分块模块分割的子块内部的句法分析和所有子块分析后的多棵子树的句法分析都是由同一个流程图完成。区别是前者的结构分析模块的 CRF 模型是用子块内部的语料训练的，所以初始输入的结点是只有词和词性的叶子结点列表，而后者的 CRF 模型是由子块间的语料训练得到的，初始输入是分析好的子树列表。

下面来具体说明结构分析模块和标签分类模块的实现。

结构分析模块。本文在结构分析模块不单需判断两个结点是否可以生产新结点，还需判断新结点的中心词是由哪个子结点继承而来，所以本文将标注集合缩减到 5 个，即：“B”、“BI”、“EI”、“E”、“O”。其中，“B”与“BI”

这两个标签是同时连续的出现，且“B”标签在前，“BI”在后，表示前词与下一个词形成一个新结点，且前词作为中心词；“EI”与“E”这两个标签也只会同时连续的出现，且“EI”标签在前，“E”标签在后，表示前词与下一个词形成一个新结点，且后词作为中心词；“O”标签代表该词不进行操作；“S”标签代表该词单独生成一个新的结点，继续以该词为中心词。

结构分析模块的 CRF 的训练语料是通过将 CTB 训练语料的子块的子树按自顶向下的顺序拆分得到，在拆分的每一步（每个结点）的过程中保留当时的特征，生成一份打标语料。这里需要注意的是拆分是按层次拆分的，每一步都是一个层的所有结点全部拆分，这样做的目的是：如果有某一层有两个结点可以拆分，反过来思考就是这两个结点的子结点可以生成两个个新结点，若我们在这一层只拆分其中一个结点，则该结点的子结点将标注为除了“O”之外的那 5 个标签之一，那么在拆分另外一个结点时，因为第一个结点的子结点此时没有不操作，即会标为“O”，这就产生了矛盾，所以本文的拆分过程是按层次的。如下图 3-10 所示，F(A)表示 A 结点的特征，B(C)表示 C 结点是 B 结点的中心成分：

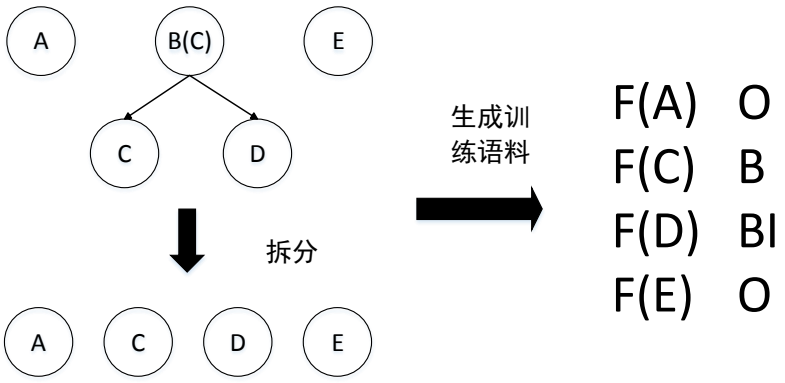

图 3-10 结构分析获得训练语料的过程

结构分析模块使用的特征列表如下表 3-2：

表 3-2 结构分析的特征列表

标记	说明
TAG	结点标签
HW	结点中心词
HP	结点中心词的词性
CLS	结点类型

其中，结点类型取值范围为 0-3。0 表示非单叉结点的中心成分为第一个子

结点；1 表示非单叉结点的中心成分非第一个子结点；2 表示结点为叶子结点；3 表示结点为单叉结点。

结构分析共四种特征，TAG 和 HP 采用的特征模板和分块模块的词特征模板是一致的，即 11 个特征组合，HW 的特征模板是：HW_{-2}、HW_{-1}、HW_0、HW_1、HW_2，CLS 的模板和 HW 一致。

训练得到模型后，利用条件随机场测试时，会得到每个结点对于 5 个标签的概率，设 $Label(Node_i)$表示结点 i 的最终标签，$P(Node_i, tag)$表示结点 i 的标签为 tag 时的概率，则结点 i 最终的标签的计算公式如下：

$$Label(Node_i) = argmax \begin{cases} P(Node_i, EI) * P(Node_{i+1}, E) \\ P(Node_i, B) * P(Node_{i+1}, BI) \\ P(Node_{i-1}, B) * P(Node_i, BI) \\ P(Node_{i-1}, EI) * P(Node_i, E) \\ P(Node_i, S) \\ P(Node_i, O) \end{cases} \quad (3\text{-}1)$$

值得注意的是在结构分析的阶段会出现两个问题：

（1） 结构分析模型测试结果全是 O，代表没有新结点产生。由于没有新结点，程序会死循环。这时程序会计算结点列表中非 O 动作概率最大的那些结点并替换它们原先的 label，进行下一步操作。

（2） 某个结点可能会产生多次 S 的动作，导致死循环。在收集可行的动作时强制限制单叉树的深度最大值为 2。

标签分类模块。在产生新结点后，新结点只是一个空结点，还未标上句法标签，标签分类模块的作用就是给予这些空结点一个合适的标签。同样的本文利用条件随机场进行标签的分类任务。具体的标签类别见附录 2 的 CTB 标注体系的短语标签表。

标签模块的 CRF 的训练语料的来源与结构分析模块是一致的，也是通过将 CTB 训练语料的子块的子树按自顶向下的顺序拆分得到，区别在于标签模块使用了更多的特征，且标签模块是利用 CRF 进行分类的任务。

标签分类模块使用的特征如下表 3-3：

表 3-3 标签分类模块的特征列表

标记	说明	提取范围
TYPE	新结点是否为二叉	—
TAG	结点标签	i-1,i,i+1,i+2
HW、HP	结点中心词和词性	i-1,i,i+1,i+2
S、S2、S3	结点结构、结点的两个子结点的标签组合	i-1,i,i+1,i+2
CLS	结点类型	i-1,i,i+1,i+2
LW、LP、RW、RP	结点中最左(右)的叶子结点的词和词性	i-1,i,i+1,i+2
NB_S1、NB_S2	结点与相邻的前后两个结点的标签的组合	—

表中提取范围的结点 i 特指产生新结点的子结点中的第一个结点，即若产生的是二叉树结点，则结点 i 指构成二叉树的最左边的子结点，若产生的是单叉树，则结点 i 指单叉树的唯一子结点。TYPE 是新结点的结点类型，所以没有提取范围。S、S2、S3 表示结点和其子结点标签的组合。标签组合是指标签字符串的联合，本文使用的是“#”，比如“NP”、“VP”和“NP”的组合是“NP#VP#NP”。LW 表示结点管辖的叶子结点中最左的叶子结点的词，同理 LP 是最左的叶子结点的词性，RW、RP 是最右的叶子结点的词和词性。NB_S1 和 NB_S2 是多个结点的标签组合，所以也没有提取范围，其中 NB_S1 表示结点 i 和结点 i-1 和 i+1 的标签组合，NB_S2 表示结点 i 和结点 i+1 和 i+2 的标签组合。

对于标签分类模块特征的模板，与结构分析模块同名的特征的模板是一致的，其余特征使用的模板与结构分析模块的 HW 的特征模板是一致的。

3.2.4 集束搜索

由于一般的 1-Best 句法分析模型，每次只保留一个最佳的句法树，容易出现句法分析早期出现错误的句法树，导致后面句法分析时的错误累加，使后面的分析变得毫无意义。所以本文引入集束搜索（Beam Search）技术。集束搜索是一种解决优化问题的一种启发式方法，本文利用集束搜索构造一个一定大小的表储存当前最优的前 K 棵树（结点列表）的 K-Best 树表，在每次进行操作时，同时对这 K 棵树进行句法分析操作，生成多棵新的树，然后从中再挑选 K 个最优树，更新目前的 K-Best 树表。依次按此操作，直至句法分析的结束。

由于标签分类的准确性达到了 98%以上，所以只有结构分析过程中使用集束搜索，而标签分类仍是 1-Best 的模式。结构分析过程获得分支的分析结果是通过，结构分析的条件随机场模型得到的序列标注。条件随机场模型可能会产生多个新结点生成的标注，本文将每一个新结点认为是一种可能的分支，并加入树表

中，具体流程如下图 3-11：

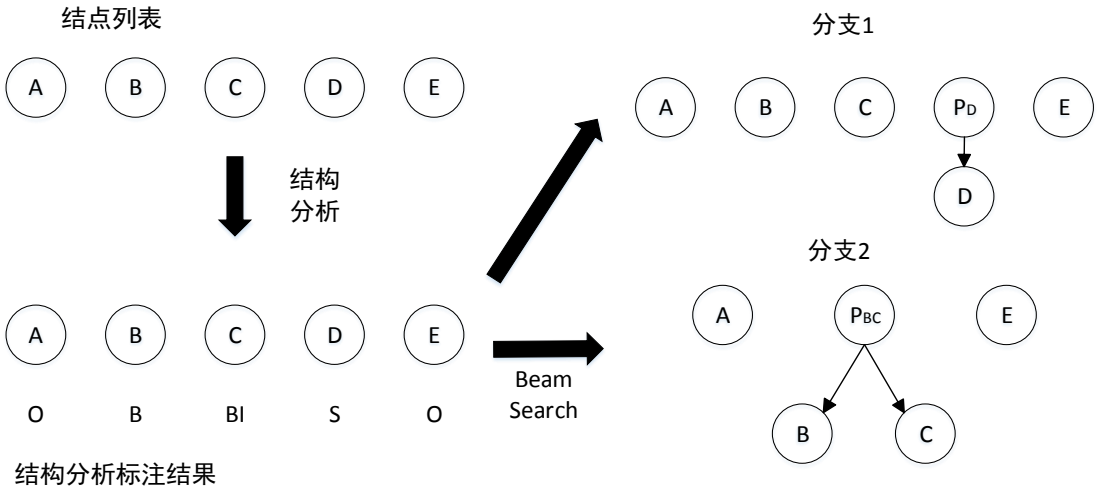

图 3-11 集束搜索演示图

集束搜索的打分机制。集束搜索在某一步获得了 N 种可能的分析结果后，本文需要对其进行打分，并选出前 K 个最佳的分支。本文采用的是当前句法树（结点）的 PCFG 的概率值作为每个分支的分数来进行比较。对于每个分支本文都通过以下的 State 保存当前的分支的结点状态和 PCFG 概率值：

$$State =< Node\_List, Score > \quad (3\text{-}2)$$

其中，$Node_List$是一个列表，存储该分支当前句法分析的状态，初始的$Node_List$是一列叶子结点列表，每个叶子结点只包含一个词和其词性；Score 存储当前句法分析的 PCFG 概率，初始 Score=0。

通过结构分析标注生成新的结点时，除了更新$Node_List$之外，Score 需加上新的规则的 PCFG 概率（采用的是对数概率，所以是加法）。举个例子，若某个分支某一时刻 n 的状态如下：

$$State_n =< [a, b, c, d], Score_n > \quad (3\text{-}3)$$

a、b、c、d 为句法分析中的结点，若此时由 b 和 c 结点产生一个新结点 p，则此时状态更新如下：

$$State_{n+1} =< [a, p, d], Score_n + P(p \rightarrow b\ c) > \quad (3\text{-}4)$$

在集束搜索中需要注意，不同状态的分支在进行完下一步新结点的生成后，可能会产生重复的分支，比如图 3-11 中，分支 1 是通过结合 B、C 结点生成的，分支 2 是通过 D 结点形成单叉树生成的，进行下一步句法分析后，可能分支 1 也发生了 D 结点形成单叉树的操作，而分支 2 可能发生 B、C 结点合并成新结点的操作，则此时，分支 1 和分支 2 的状态是一模一样的。对于重复的分支，系统

需要识别，并及时删除，避免集束搜索中的树表中出现 k 棵树都一模一样的错误。

3.3 实验与分析

本节将对上文提到的句子分块、结构分析模块、标签分类模块以及整体的句法分析的效果分别设计了四个实验。

3.3.1 实验数据介绍

本文使用了 CTB8.0 树库[51]作为句法分析的语料的树库。CTB8.0 是由语言数据协会于 2013 年发布的树库，其属于短语结构树库，是宾州中文树库的发展。CTB8.0 包含 3007 个文本文件，71369 句，1620561 词，2589848 个字，包含网络采集语料、通讯新闻、新闻杂志、广播新闻、广播谈话、博客等类型语料，其标注体系见附录 1 与附录 2。具体的语料分布如下表 3-4：

表 3-4 CTB8.0 树库中的文件类型分布

体裁类型	文件分布(id)
Newswire	[0001-0325, 0400-0454, 0500-0540, 0600-0885, 0900-0931, 4000-4050]
Magazine articles	[0590-0596, 1001-1151]
Broadcast news	[2000-3145, 4051-4111]
Broadcast conversations	[4112-4197]
Weblogs	[4198-4411]
Discussion forums	[5000-5558]

本文选取了 Newswire 和 Magazine articles 作为本文的句法分析语料，原因是 Broadcast news、Broadcast conversations、Weblogs 和 Discussion forums 的体裁更加多样，跨度更加广，为了避免受此影响，所以选择体裁和文风较为稳定和用词较为规范的 Newswire 和 Magazine articles。其中因为本文的句法分析系统没有超参数，所以本文只设置了测试和训练两个集合，其文件分布如下表 3-5：

表 3-5 CTB8.0 树库中训练和测试文件分布

语料集	体裁类型	文件分布(id)	句子总数	平均词数
训练集	Newswire	[0001-0325, 0400-0454, 0500-0540, 0600-0885, 0900-0931]	18778	28.07
	Magazine articles	[0590-0596, 1001-1151]		
测试集	Newswire	[4000-4050]	326	28.12

其中测试集句子的长度分布如下图 3-12：

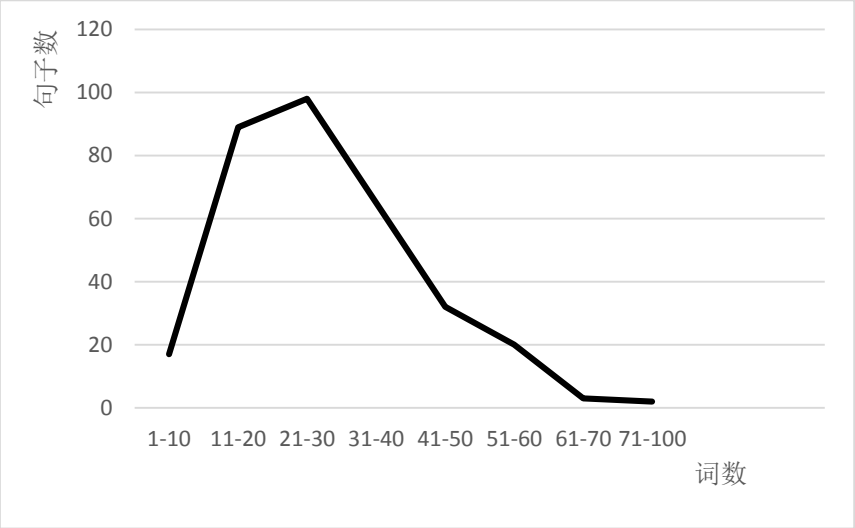

图 3-12 测试集句子的长度分布

可以看出测试集中的句子的长句主要集中在 11-40 这个区间共 252 句，大于 40 词的句子仍有 57 句，所以对于句法分析在长句的性能是一个很大的挑战。

特别注意的是，在 CTB8.0 中存在一些繁体字，如“*於*”、“*麼*”、“*著*”（通“*着*”）、“*後*”、“*什麼*”等，所以在使用 CTB8.0 语料的时候需要将其转化成简体字。

3.3.2 实验分析

3.3.2.1 句子分块实验

本小节将讨论句子分块的实验效果。分块的评价方法有两种。其中一种分块的评价方法本文采用是准确率（Precision）和召回率（Recall）以及准确率与召

回率的调和平均数 F-测度（F1），它们的定义如下：

$$P = \frac{召回正确的子块数}{系统划分的总子块数} \tag{3-5}$$

$$R = \frac{召回正确的子块数}{正确的总子块数} \tag{3-6}$$

这里“*正确的子块*”表示的是与正确标注完全一致的子块。F1 的定义如下：

$$F1 = \frac{2 * P * R}{P + R} \tag{3-7}$$

实验数据的设置在 3.2.2 已经介绍，将训练集中的句法树按分块的准则拆分成子块，并标上序列标注，这里采用的是“B”、“B_2”、“M”、“E_2”、“E”、“S”这六个标签，具体结果如下：

表 3-6 分块 PRF 评价体系的结果

准确率	召回率	F1
0.7756	0.8707	0.8204

第二种评价方法是直接计算系统得到的分块中“*没有错误的子块*”的准确率，这里的“*没有错误的子块*”的定义是：如果系统得到的子块满足，可以在正确的句法树中找到某个结点，其包含且仅包含子块中所有的词，则我们称这个子块是“*没有错误的子块*”。因为该子块可能不属于按照分块准则拆分的子块集合中，但是该子块不会产生跨结点的边界冲突，所以不会影响后续的句法分析。第二种评价方法的计算公式如下：

$$P = \frac{没有错误的子块数}{系统划分的总子块数} \tag{3-8}$$

第二种分块评价方法的结果是：0.916。

观察两种分块的评价方法，可以看出，分块作为整个句法分析的第一个步骤，其性能不是很理想，可能会影响系统的整体效果，未来可能需要进一步改进。

3.3.2.2 结构分析实验

本小节将讨论结构分析模块的标注效果。针对结构分析的评价方法，本文也是采用准确率（P）和召回率（R）以及 F-测度，它们的定义如下：

$$P = \frac{召回的正确的操作数}{系统的总操作数} \tag{3-9}$$

$$R = \frac{召回的正确的操作数}{正确的总操作数} \tag{3-10}$$

这里的操作包括“B”、“E”、“S”、“O”，分别表示“B”与“BI”结点结合成新结点、“EI”与“E”结点结合成新结点、“S”结点单独生成一个结点、“O”结点不做操作。对于 F1 的计算公式同式 3-7。

结构分析的实验分为两个部分，一个是子块内部的结构分析实验，另一个是子块间的结构分析实验。训练语料的产出在 3.2.3 的结构分析中已经介绍，这两者的区别是前者的训练数据是通过拆分子块的子树至叶子结点得到的，后者是通过拆分完整的句法树至子块的子树的过程得到的。两者的具体实验结果如下：

表 3-7 子块内的结构分析的实验结果

类型	数量	P	R	F1
B	1394	0.8647	0.9261	0.8944
E	5247	0.8782	0.8948	0.8864
S	3807	0.9490	0.8789	0.9126
O	13904	0.9384	0.9307	0.9345
标签	30993	0.9117	0.9117	0.9117
所有操作	10448	0.9003	0.8932	0.8967

表 3-8 子块间的结构分析的实验结果

类型	数量	P	R	F1
B	174	0.9144	0.7988	0.8527
E	2027	0.9056	0.9427	0.9238
S	28	0.8235	0.5000	0.6222
O	6920	0.9720	0.9565	0.9642
标签	11350	0.9456	0.9456	0.9456
所有操作	2229	0.9056	0.9259	0.9157

实验结果中的“B”、“E”、“S”、“O”是操作，标签指序列标注的标签的性能，所有操作指“B”、“E”、“S”、“O”四个操作的整体性能。从

PRF 的评价结果分析，我们可以观察到整体上子块间的结构分析的效果要优于子块内的效果，造成这个结果的原因应该是进行子块间结构分析时，其特征会更加丰富，而子块内部结构分析时有很多的特征为空。此外，子块间的“S”操作的准确率较低，可能是由于数据量太少（只有 28 个），造成的偏差过大。从数量上分析，我们可以看到在子块内部和子块间的最多的操作是“E”，说明大多数的中心成分位于子结点右侧；在子块内部操作第二多的操作是“S”，说明子块内部的单叉树较多。

3.3.2.3 标签分类实验

本小节将讨论标签分类模块的标注效果，以及不同的特征的选取对标签分类的影响。针对标签分类的评价方法，本文采用的是准确率（P）的评价方法，定义如下：

$$P = \frac{召回的正确的标签数}{总标签数} \tag{3-11}$$

这里的标签不包括词性（默认输出是带词性的词序列）。训练语料和测试语料的产出在 3.2.3 的标签分类中已经介绍。标签分类的实验不分块内标签分类和块间标签，统一一起进行实验，这里总共分为两个子实验，一个是利用 Word2Vec 转化频率较小的词为频率较大的相似词，另一个实验室考察特征选择的实验。

本文使用了 Word2Vec，并利用整个 CTB8.0 语料和人民日报语料训练了一个 50 维词向量模型，Word2Vec 运用在标签分类的实验结果如下：

表 3-9 标签分类的 Word2Vec 词转化实验结果

实验设置	P
Base	0.9770
All_transfer	0.9761
NVA_transfer	0.9762

表中，Base 是使用了 TYPE、TAG、HW、HP、CLS、LW、LP、RW、RP 特征的标签分类系统。All_transfer 指将输入特征中的词频低于 5 的词全部利用 Word2Vec 查找相似度最高的词替换之。NVA_transfer 指只将输入特征中的词频低于 5 且词性是名词、动词和形容词（包括副词）替换成 Word2Vec 中相似度最高的词。

可以从表中看出，Word2Vec 的加入没有提高标签分类的性能，一个原因可

能是因为 CTB 的语料中低频词较少，另外还可能是因为 Word2Vec 的模型的训练语料太少只有不到 100M 的规模。

特征选取的实验结果如下：

表 3-10 标签分类的特征选取实验结果

特征选取	P
Base	0.9770
Base+S	0.9804
Base+S+NB_S	0.9842
Base+S+NUM	0.9807
Base+S+LEN	0.9806
Base+S+NB_S+ NUM+ LEN	0.9840

表 3-10 中 S 指表 3-3 中的 S、S1、S2，NB_S 指表 3-3 中的 NB_S1 和 NB_S2 特征，NUM 指结点中 NP 和 VP 的数量的特征，LEN 指结点中所有词列表的长度。

从表中可以看出组合特征 S 和 NB_S 大大提高了标签分类的性能，NUM 和 LEN 特征对标签分类性能的提高不如 NB_S。句法分析系统最后选取的特征是 Base+S+NB_S。

3.3.2.4 句法分析整体实验

本小节将讨论句子分块与集束搜索对句法分析整体的性能的影响。本文采用的是 PARSEVAL 句法分析的评价体系，主要包含准确率（P）、召回率（R）和 F1 三个指标，其定义如下：

$$P = \frac{分析器剖析出的正确短语个数}{分析器剖析出的短语个数} \quad (3\text{-}12)$$

$$R = \frac{分析器剖析出的正确短语个数}{标准句法树中的短语个数} \quad (3\text{-}13)$$

F1 的定义同式 3-7。其中的短语的定义是指句法树中所有非叶子结点。在评价时使用该短语的标记和短语管辖的词的范围表示短语，所以只有当短语的标签和管辖的词与标注句法树中的短语完全一致时，该短语才被认为是一个正确的短语。举个例子，图 1-1 中的句法树在评价时的短语有：(ADVP, [0,1])、(VP, [0,6])、

(VP, [1,6])、(NP, [2,6])、(NP, [2,4])、(NP, [4,6])、(PU, [6,7])、(IP,[0,7])。[n,m]表示第 n 个词至第 m-1 个词的范围。

实验的结果如下表 3-11：

表 3-11 句法分析实验结果

系统	P	R	F1
Stanford_pcfg	0.6422	0.6190	0.6304
Stanford_factored	0.6783	0.6631	0.6706
Berkeley_pcfg	**0.8054**	**0.7839**	**0.7945**
Multiple	0.7725	0.7450	0.7585
Beam=1	0.7379	0.6743	0.7047
Beam=2	0.7466	0.6925	0.7186
Beam=4	0.7554	0.7036	0.7286
Beam=8	0.7580	0.7067	0.7315
Beam=16	0.7543	0.7008	0.7266

表中，Stanford_pcfg 和 Stanford_factored 都是斯坦福大学自然语言处理小组开发的句法分析工具，前者是运用 Parent annotation 和 RHS markovization 的 PCFG 算法的模型，后者是以 PCFG 和 DEP 为基础模型，词汇化的 A*算法的句法分析模型。Berkeley_pcfg 是加州大学伯克利分校自然语言处理实验室开发的一种基于 PCFG 的句法分析器。除上述三个句法分析系统外，另外 6 个是本文的句法分析系统的在不同设置情况的实验结果。为了和这些主流的句法分析模型进行横向的比较，这些系统都是在相同的训练语料上得来的，实验结果也是在相同的测试语料得到的。

从表中可以看出，本文的 6 个系统都超过了 Stanford 的句法分析系统，但是低于 Berkeley 的句法分析系统的 F1=0.7945。此外，引入集束搜索的句法分析，随着 Beam 的增大，句法分析效果逐渐提高，并在 Beam=8 处达到最好，当 Beam=16，句法分析效果有所下降。但是集束搜索的模型取得的最佳结果低于 Multiple 的结果，前者的 F 值最高为 Beam=8，达到了 0.7315，而 Multiple 模型达到了 0.7585。Multiple 的设置与集束搜索有所不同，在标记结构分析过程中，条件随机场可能会同时显示多个新结点的产生，如图 3-11，此时集束搜索会分别产生两个分支，而 Multiple 则会同时进行两个结点的生成，所以 Multiple 的句法分析模型会比引入集束搜索的模型快（Multiple 模型跑完 327 句测试语料花费 544

秒，而 Beam=8 的模型花费 788 秒），在这里其性能也比集束搜索模型略优。

3.4 本章小结

本章主要讨论了本文的句法分析系统的具体实现。本文借鉴了层次化汉语句法分析的两级句法分析的方法，首先将句子分割成几个子句，然后对子句进行句法分析操作，生成一棵子树，接着对每个子树进行合并操作，最后形成一个完整的句法树。在句法分析过程中，为了表达更加丰富的特征信息，本文引入中心驱动模型的特点——中心成分，对每个结点都标注了它的中心成分，并利用移近归约依存句法分析模型的思想，将句法分析问题转化成分类问题，并进一步转化成序列标注问题，利用条件随机场模型训练一个序列标注模型。在句法分析的整体实现中，本文采用集束搜索技术大大缓解句法分析早期的错误累加问题，进一步提高了句法分析性能。最后本文设计了几个实验证明了，本文的句法分析模型的可行性，为下一章基于句法分析的旅游信息检索系统提供了理论基础。

第四章 旅游信息检索系统

信息检索是指根据用户的需求，从按一定组织的信息集合中查找出相关的信息的过程。词袋模型（Bag of words, BOW）是目前信息检索最普遍的采用的模型，但是该模型只是基于简单的词项匹配和相似度计算，不能深层挖掘用户的检索意图。自然语言检索（natural language Retrieval, NLR）是信息检索的一种，其能够有效的对自然语言处理进行多个层次的分析，使机器能真正理解用户的检索需求。本章主要介绍本文实现的一个利用句法分析提取关键词的旅游信息检索系统。

4.1 系统概述

本文实现了旅游信息检索系统，该系统存在一个结构化的旅游信息知识库，能够根据用户的检索准确的返回用户需求的信息。

旅游信息检索系统的使用的环境是 Ubuntu，编程语言是 Python2.7，整个检索系统包括如下几个模块，具体流程图见图 4-1：

1）系统界面。使用 Tkinter 开发，Tkinter 是 Python 提供的图形用户界面模块，可以跨平台使用。具体效果可以见图 4-2 和图 4-3。
2）分词和词性标注模块。该模块主要利用中科院的 NLPIR 分词工具包和开源项目 ctbparser-master 工具包完成。
3）句法分析模块。该模块是检索系统的关键技术，通过句法分析能够深层次地挖掘用户的需求，从而提高检索的准确率。
4）关键词提取模块。该模块是检索系统的核心部分，主要作用是利用用户搜索词条的词性标注和句法分析的结果提取一个六元组，包括实体、疑问词、名词、动词、时间词等信息。
5）问题分类模块。问题分类模块是通过分词结果以及其扩展的同义词构造词的空间向量模型，并利用支持向量机对用户的词条进行分类。
6）知识库匹配模块。该模块利用关键词提取结果和问题分类的结果，从系统的知识库中返回匹配的信息。

知识库是由多个三元组构成的数据结构。例如下面这个三元组：“*（天安门&高度，8，天安门城楼原高 33.7 米,1970 年 2 月重修后,为 34.7 米）*”。其中第一个成分是由实体（Entity）和实体属性组成，类似哈希结构中的键的作用；第二个成分是一个数字，代表该三元组对应的信息类别，在检索过程中需要与查询

句的类别相对应；最后一个成分是由第一个成分所对应的信息，类似哈希结构中的键对应的值。所以当我们根据用户检索得到天安门和高度这些关键词且用户的问题类别也是 8，则我们需要返回其对应的值，即“*天安门城楼原高 33.7 米,1970 年 2 月重修后,为 34.7 米*”。

系统使用了同义词词典，该词典主要由哈工大同义词林经过修改得来。系统主要在关键词扩展和问题分类模块中使用同义词典。该同义词词典结构比较简单，默认一行中的所有词为同义词。

问题的类别共 9 种，包括：其他、介绍、人物、位置、时间、天气、价格、尺寸和周边。具体含义见附录 5。特别地，当问题类别为“其他”时，系统将出现检索失败，认为用户的查询词条不能从知识库中检索到有效信息。

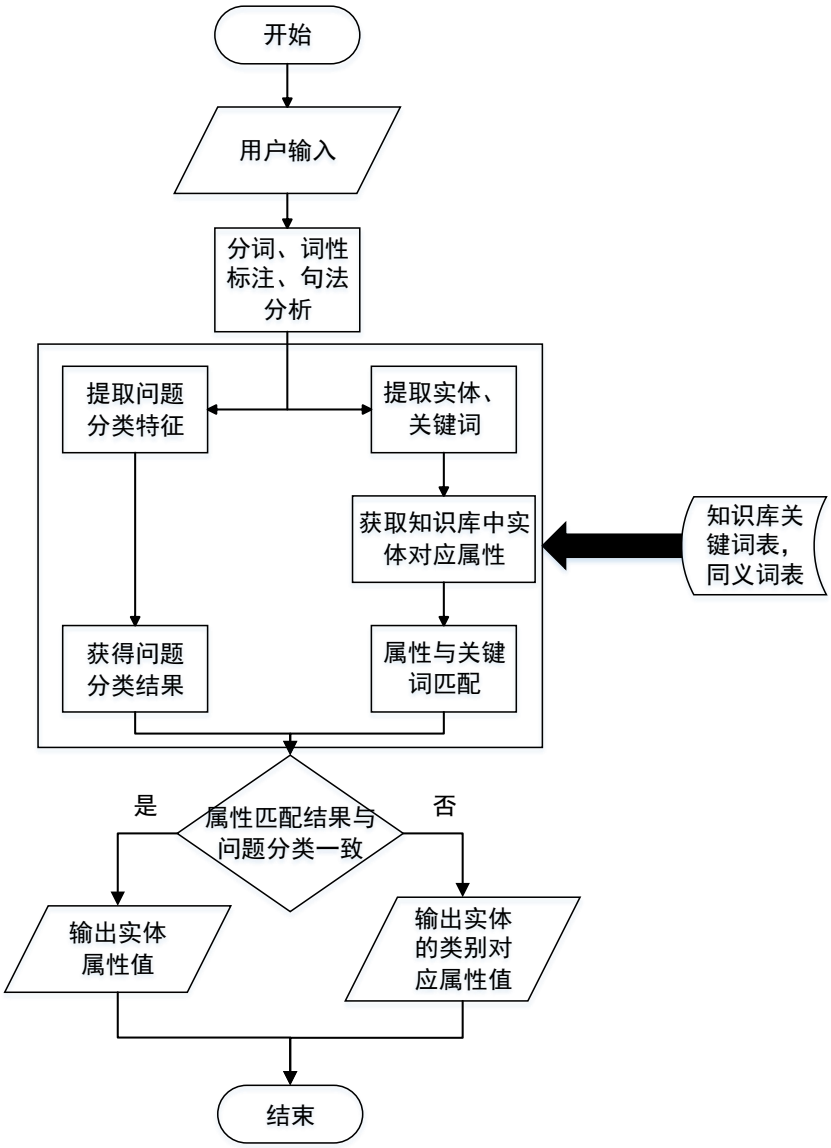

图 4-1 旅游信息检索系统流程图

4.2 系统运行效果

旅游信息检索系统的界面主要包括几个部分：用户输入框、清空和搜索按钮、检索结果显示框、句法树结果显示框以及显示句法树图片按钮。具体运行效果见下图 4-2 和图 4-3。图 4-2 为系统初始界面，即用户未检索时的界面。图 4-3 为用户输入“*天安门国旗有多高*”时系统的运行效果，检索结果显示的第一行是系统提取的关键词，可以看出“*高*”已经被同义扩展为“*高度*”，第二行是问题的类别，图中显示该问题类别是尺寸类，第三行是检索的结果，右边为句法分析结果。

图 4-2 旅游信息检索系统界面

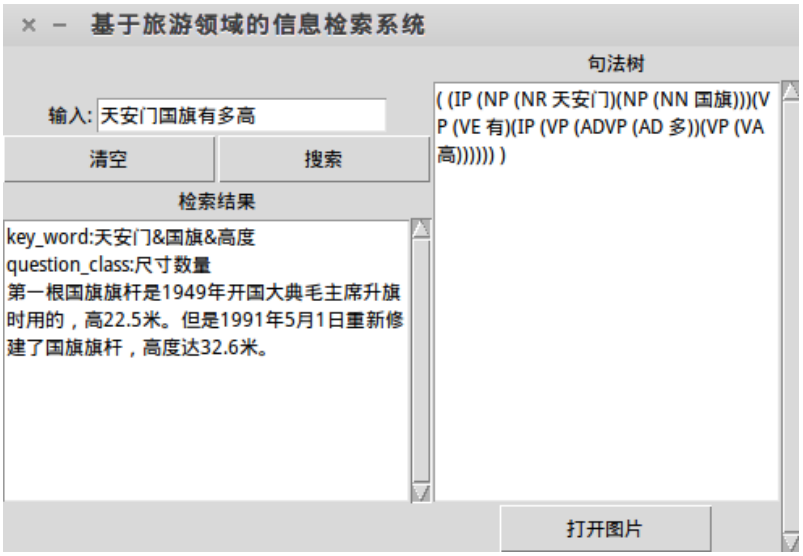

图 4-3 旅游信息检索系统运行结果示例图

4.3 实验与分析

旅游信息检索系统的实验主要对比使用规则和使用句法分析的关键词提取的命中情况。

语料来源。本文利用爬虫在网络上的一些问答页面爬取一些问题，主要是根据关键词爬取，关键词主要是旅游领域的一些实体，比如：天安门、故宫、人民英雄纪念碑等名词。最后经过修改，总共 539 句，其中字数大于 15 的句子共 103 句。规则方法和句法分析方法的实验结果如下：

表 4-1 旅游信息检索系统实验结果

	句法分析方法正确	句法分析方法错误
规则方法正确	464	16
规则方法错误	42	17

表 4-2 旅游信息检索系统实验结果(字数大于 15 的问句)

	句法分析方法正确	句法分析方法错误
规则方法正确	64	7
规则方法错误	25	7

从实验结果中可以看出，总体上基于句法分析的方法的准确率是 93.8%，而基于规则的方法是 89.1%，但是在字数大于 15 的句子试验中，前者准确率依然有 86.4%，而后者准确率降至 68.9%。所以我们可以得出以下结论：基于句法分析的方法相较于基于规则的方法更好，且随着问句的字数增加，其性能不会迅速下降。该实验也表明，本文实现的基于分块的句法分析在具体任务中也能有较好的表现。

下面举几个具体的例子：

比如，在检索“*去毛主席纪念堂不能带什么去啊?*”时，规则方法只提取到了“*去*”作为动词关键词，而句法分析方法则提取出“*带*”这个最关键的关键词。

再如，在检索“*人民英雄纪念碑碑体侧面图像是什么?什么含义?*”时，规则方法提取到了“*碑*”和“*含义*”作为名词关键词，句法分析方法提取出“*侧面图*”和“*含义*”，显然“*侧面图*”这个关键词更为重要。

这些例子体现了，句法分析方法能够提取出规则方法所不能挖掘的更为重要的关键词。

4.4 本章小结

本章主要介绍引入了句法分析的基于旅游领域的信息检索系统。利用句法分析，我们可以得到句子的句法结构以及其中的中心成分信息，这些信息可以帮助系统更便捷的得到检索词条的关键词。本章通过介绍系统的主要模块以及与基于规则的检索的比较，体现出句法分析在信息检索中的重要作用。

第五章 总结与展望

5.1 本文工作总结

句法分析是是自然语言处理研究中的关键技术之一。句法分析是信息检索、机器翻译、情感分析等上层自然语言任务的基础，其性能的每一步提高都会大大促进这些任务的发展。

由于汉语表意的特性导致汉语句子结构较为松散，使得汉语句法分析技术的发展遇到很大的瓶颈，尤其是长句的句法分析是目前国内外句法分析的研究者都想要努力攻克的一个难题，也是未来句法分析的发展的主要方向之一。本文借鉴了层次化汉语句法分析模型“分而治之”的思想，将整个句法分析流程拆分成两级处理，第一级是分块模块，目的是将长句分割成较短的子块；第二级是句法分析模块，负责具体的句法分析，包括新结点的生成和新结点的标签打标。这两级处理的具体内容如下：

分块模块。采用的不是直接利用分割符号分割句子，而是首先利用分割符号将训练集中的句法树分成几个子结点，确保子结点内部不包含分割符号，然后将子树转成子块，并用条件随机场模型学习这种分块方法深层的规律，构造一个分块的模块。

句法分析模块。借用了基于移近归约的依存句法分析模型的思路，将句法分析转化成序列标注的结构分析和标签分类两个子模块，同样利用条件随机场模型完成两个子模块的实现。利用基于中心驱动模型的句法分析的精髓，引入中心词信息，大大丰富了句法分析模块的特征体系，从试验结果中也能看出，中心词信息可以大大提高了句法分析的性能。

通过实验，将本文提出的句法分析模型和目前主流的句法分析工具包 Stanford Parser 和 Berkeley Parser 进行了比较，在同样的训练语料和测试语料下，本文的句法分析的结果要优于 Stanford Parser 的两个模型，但是略低于 Berkeley Parser 的模型。在模型的内部比较试验中，我们发现引入集束搜索技术会提高句法分析的性能，本文的句法分析模型在 Beam 取 8 时取得较好的结果。本文最好的结果是 Multiple 模型，该模型的特殊之处在于其采用的是一次可以形成多个新结点的方法，这说明句法分析的过程可能不是简单地一个结点接着一个结点的生成的过程。

最后，本文将句法分析模型应用到了基于旅游领域的信息检索系统。句法树的中心成分的信息可以有效提取检索词条的核心词汇，相较于基于规则的抽取方法大大改善了信息抽取的性能。

5.2 未来研究工作

虽然本文提出的基于分块的汉语句法分析模型取得了一定的效果，但是从试验结果中可以看出其中仍然存在一些问题，导致系统的整体性能仍低于一些PCFG 的模型。

1）分块模块的整体性能不高，大大影响了句法分析的效果；

2）对于新词或者低频词，系统不能较好地识别和处理，导致分析结果容易出现错误；

3）本文较为依赖条件随机场模型，但是本文利用的是 CRF++工具的命令行操作，通过文本的读写来读取模型的输出结果，这是系统主要的耗时原因。

针对以上问题，在未来的工作中可以从以下几个方面进行改进：

首先，针对分块性能较低的问题，可以考虑优化分块的特征和参数，比如设定分块的大小或者调整分块的条件随机场模型的标签设置；对于新词或者低频词的问题，我们可以借助同义词典对这些词进行转换，也可以引入大规模语料训练的 Word2Vec 或 LDA 模型，将词特征转化为向量特征，降低新词对系统的影响程度；对于条件随机场模型耗时的问题，我们可以仔细研究条件随机场模型，尝试将其转入内存运行并直接读取模型的输出；对于集束搜索，我们目前是采用 PCFG 的概率值作为分数来选取最佳的分支，未来可以考虑基于中心词的 PCFG 概率或者其他方法计算分数。

其次，显然句法分析不是简单的基于机器学习的方法就能轻易解决的难题，我们需要深入研究汉语语言学，了解汉语深层次的语义，学习语言的本质、结构和发展规律。深入了解了汉语的特性后，我们或许会发现一些汉语句法的一些特性，从而得到一些建设性的改进点。

最后，句法分析作为自然语言处理的核心任务，经历了几代研究者的锲而不舍的努力，才得到了今天的成果。虽然句法分析研究仍旧有些问题得不到解决，但是在科学和技术日益更新的时代，我相信总有一天汉语句法分析会取到突破性的进展，我期待着这一天的到来。

参考文献

[1] Jurafsky D, Martin J H. Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition[M]// Pearson/Prentice Hall, 2009:638-641.

[2] 刘鸿宇，赵妍妍，秦兵,等. 评价对象抽取及其倾向性分析[J]. 中文信息学报，2010, 24(1):84-88.

[3] Lucy Vanderwende, NLPwin – an introduction, no. MSR-TR-2015-23, March 2015.

[4] Magerman D. Statistical decision-tree models for parsing[J]. Meeting of the Association for Computational Linguistics, 1995.

[5] Marcus M P, Santorini B, Marcinkiewicz M A. Building a large annotated corpus of English: The Penn Treebank[C]// Computatzonal Linguistics. 1993:313-330.

[6] Ratnaparkhi A. Learning to Parse Natural Language with Maximum Entropy Models[J]. Machine Learning, 1999, 34(1-3):151-175.

[7] Collins M J, Marcus M P. Head-Driven Statistical Models for Natural Language Parsing.[J]. Computational Linguistics, 2003, 29(4):589-637.

[8] Collins M. Discriminative Reranking for Natural Language Parsing[J]. Computational Linguistics, 2005, 31(1):25-70.

[9] Klein D, Manning C D. Accurate unlexicalized parsing[C]// Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. Association for Computational Linguistics, 2003:423-430.

[10] Henderson J. Neural Network Probability Estimation for Broad Coverage Parsing.[C]// Conference of the European Chapter of the Association for Computational L. 2003:131--138.

[11] Petrov S, Barrett L, Thibaux R, et al. Learning Accurate, Compact, and Interpretable Tree Annotation[J]. Acl ', 2006:433--440.

[12] Matsuzaki T, Miyao Y, Tsujii J. Probabilistic CFG with latent annotations[C]//Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005: 75-82.

[13] Shindo H, Miyao Y, Fujino A, et al. Bayesian symbol-refined tree substitution grammars for syntactic parsing[C]// Annual Meeting of the Association for Computational Linguistics. 2012:440-448.

[14] Hall D, Durrett G, Dan K. Less Grammar, More Features[J]. Eecs.berkeley.edu, 2014:228-237.

[15] Tomita M. Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems[J]., 1985.

[16] Tomita M. A Probabilistic Parsing Method for Sentence Disambiguation[M]// Current Issues in Parsing Technology. Springer US, 1991:139-144.

[17] Daniel H. Younger. Recognition and parsing of context-free languages in time n 3 *[J]. Information & Control, 1967, 10(2):189-208.

[18] Kasami T. An efficient recognition and syntax analysis algorithm for context free languages[J]., 1965.

[19] Earley J. An efficient context-free parsing algorithm.[J]. Communications of the Acm, 1970, 13(2):94-102.

[20] Leech G, Garside R. Running a grammar factory: the production of syntactically analysed corpora or 'treebanks[J]. in: Stig Johansson and Anna-Brita Stenstrom: English Computer Corpora: Selected Papers and Research Guide, Mouton de Gruyter, 1991.

[21] Marcus M P, Marcinkiewicz M A, Santorini B. Building a large annotated corpus of English: The Penn Treebank[J]. Computational linguistics, 1993, 19(2): 313-330.

[22] Bikel D M, Chiang D. Two Statistical Parsing Models Applied to the Chinese Treebank[C]// In Proceedings of the Second Chinese Language Processing Workshop. 2000:1--6.

[23] 周强, 黄昌宁. 基于局部优先的汉语句法分析方法[J]. 软件学报, 1999, 10(01):1-6.

[24] 张玥杰, 朱靖波, 张跃,等. 基于 DOP 的汉语句法分析技术[J]. 中文信息学报, 2000, 14(01).

[25] Luo X. Q.. A maximum entropy Chinese character-based parser. In Proceeding of theConferenceon Empirical Methods in Natural Language Porcessing.2003. 192-199.

[26] 吕雅娟, 李生, 赵铁军. 基于双语模型的汉语句法分析知识自动获取[J]. 计算机学报, 2003, 26(1):32-38.

[27] 曹海龙. 基于词汇化统计模型的汉语句法分析研究[D]. 哈尔滨工业大学, 2006.

[28] 李军辉, 周国栋, 朱巧明,等. 基于层次模型的中文句法分析[C]// 第三届全国信息检索与内容安全学术会议论文集. 2007.

[29] Wang M.W., Sagae K., Mitamura T.. A Fast, Accurate Deterministic Parser for Chinese. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. 2006. 425-432.

[30] 袁里驰. 基于词聚类的依存句法分析[J]. 中南大学学报：自然科学版, 2011, 42(07):2023-2027.

[31] Qian X, Liu Y. Joint Chinese word segmentation, POS tagging and parsing[C]//

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012:501-511.

[32] 段慧明, 徐国伟, 胡国昕, 等. 大规模汉语标注语料库的制作与使用[J]. 语言文字应用, 2005 (2): 72-77.

[33] Xue, Nianwen, et al. Chinese Treebank 8.0 LDC2013T21. Web Download. Philadelphia: Linguistic Data Consortium, 2013.

[34] 周强. 汉语句法树库标注体系[J]. 中文信息学报, 2004, 18(04):1-8.

[35] 杨沐昀. 汉英句子对齐及翻译词典和翻译规则的自动获取[D]. 哈尔滨工业大学, 2002.

[36] Liu T, Ma J, Li S. Building a Dependency Treebank for Improving Chinese Parser[J]. Journal of Chinese Language and Computing, 2006, 16(4): 207-224.

[37] Chomsky N. Rules and representations[J]. Behavioral & Brain Sciences, 1980, 3(1):1-15.

[38] Xia F. Automatic grammar generation from two different perspectives[D]. University of Pennsylvania, 2001.

[39] Li X, Zong C, Hu R. A Hierarchical Parsing Approach with Punctuation Processing for Long Sentence Sentences[C]//In Proceedings of the Second International Joint Conference on Natural Language Processing: Companion Volume including Posters/Demos and Tutorial Abstracts. 2005.

[40] 李幸, 宗成庆. 引入标点处理的层次化汉语长句句法分析方法 12[J]. 2006.

[41] 1995 G B T. 标点符号用法[D]. , 1995.

[42] Gaifman H. The equivalence of context-free phrase structure grammars and categorical grammars[r]. Hebrew Univ Jerusalem (Israel), 1965

[43] Yamada H, Matsumoto Y. Statistical dependency analysis with support vector machines[C]//Proceedings of IWPT. 2003, 3: 195-206.

[44] Lafferty J, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[J]. 2001.

[45] 李航. 统计学习方法[J]. 2012.

[46] Kudo T. CRF++: Yet another CRF toolkit[J]. Software available at http://crfpp. sourceforge. net, 2005.

[47] QIAN X. Pocket CRF[J]. 2008-08-05)[2009-02-25]. http://sourceforgenet/projects/pocket-crf-1/files.

[48] Okazaki N. CRFsuite: a fast implementation of conditional random fields (CRFs)[J]. URL http://www. chokkan. org/software/crfsuite, 2007.

[49] Chomsky N. Remarks on Nominalizationl[J]. 1968.

[50] Nunberg G. The linguistics of punctuation[M]. Center for the Study of Language (CSLI), 1990.

[51] Xue, Nianwen, et al. Chinese Treebank 8.0 LDC2013T21. Web Download. Philadelphia: Linguistic Data Consortium, 2013.

附录

附录 1 CTB 语料库词性标记

标记	英语解释	中文解释
AD	adverbs	副词
AS	Aspect marker	体态词，体标记，例子：了、在、着、过）
BA	把 in ba-const	“把”，“将”的词性标记
CC	Coordinating conjunction	并列连词，“和”
CD	Cardinal numbers	数字，例子：“一百”
CS	Subordinating conj	从属连词，例子：“若”、“如果”、“如”
DEC	的 for relative-clause etc	“的”词性标记
DEG	Associative 的	联结词“的”
DER	得 in V-de construction, and V-de-R	“得”
DEV	地 before VP	“地”
DT	Determiner	限定词，例子：“这”
ETC	Tag for words 等，等等 in coordination phrase	“等”、“等等”
FW	Foreign words	例子：ISO
IJ	interjection	感叹词
JJ	Noun-modifier other than nouns	名词修饰词，例子：“传统”
LB	被 in long bei-construction	例子：“被”、“给”
LC	Localizer	定位词，例子：“里”
M	Measure word (including classifiers)	量词，例子：“个”
MSP	Some particles	例子：“所”
NN	Common nouns	普通名词
NR	Proper nouns	专有名词
NT	Temporal nouns	时序词，表示时间的名词
OD	Ordinal numbers	序数词，例子：“第一”
ON	Onomatopoeia	拟声词，例子：“哈哈”
P	Prepositions (excluding 把 and 被)	介词

（续上表）

标记	英语解释	中文解释
PN	pronouns	代词
PU	Punctuations	标点
SB	被 in long bei-construction	例子：“被”、“给”
SP	Sentence-final particle	句尾小品词，例子：“吗”
VA	Predicative adjective	表语形容词，例子：“红”
VC	Copula 是	系动词，例子：“是”
VE	有 as the main verb	例子：“有”
VV	Other verbs	其他动词

附录 2 CTB 语料库短语标记

标记	英语解释	中文解释
ADJP	Adjective phrase	形容词短语
ADVP	Adverbial phrase headed by AD (adverb)	由副词开头的副词短语
CLP	Classifier phrase	量词短语
CP	Clause headed by C (complementizer)	由补语引导的补语从句
DNP	Phrase formed by “XP+DEG”	名词修饰短语
DP	Determiner phrase	限定词短语
DVP	Phrase formed by “XP+DEV”	动词构成副词短语
FRAG	fragment	标记
IP	Simple clause headed by I (INFL)	简单句
LCP	Phrase formed by “XP+LC”	LC 位置词
LST	List marker	列表标记，如“--”
NP	Noun phrase	名词短语
PP	Preposition phrase	介词短语
PRN	Parenthetical	括号中的，插入的
QP	Quantifier phrase	量词短语
UCP	unidentical coordination phrase	非对等同位语短语
VP	Verb phrase	动词短语
VCD	Coordinated verb compound	并列动词复合
VCP	Verb compounds formed by VV+VC	动词+是
VNV	Verb compounds formed by A-not-A or A-one-A	动词+否定词（量词）+动词
VPT	Potential form V-de-R or V-bu-R	V-de-R, V 不 R
VRD	Verb resultative compound	动词结果复合
VSB	Verb compounds formed by a modifier + a head	定语+中心词

附录 3 不同结构的推导树总数随句子长度增长变化表

句子长度(词)	推导树数量
1	1
2	1
3	2
4	5
5	14
6	42
7	132
8	429
…	…
20	1767263190
21	6564120420
…	…

附录 4 Xia F 论文中汉语中心词规则表

成分标记	方向	优先中心词标记列表
ADJP	right	ADJP/JJ
ADVP	right	ADVP/AD
CP	right	CP/IP
DNP	right	DNP/DEG
DP	left	DP/DT
INTJ	left	INTJ/JJ
IP	right	IP/VP
LCP	right	LCP/LC
NP	right	NP/NN/NT/NR/QP
PP	left	PP/P
QP	right	QP/CD/OD
VP	right	VP/VA/VC/VE/VV/BA/LB/VCD/VSB/VRD/ VNV/VCP
VV	right	VV
VA	right	VA
VE	right	VE
VC	right	VC
VCD	right	VCD/VV/VA/VC/VE
VRD	right	VRD/VV/VA/VC/VE
VSB	right	VSB/VV/VA/VC/VE
VCP	right	VCP/VV/VA/VC/VE
VNV	right	VNV/VV/VA/VC/VE

附录 5 旅游信息检索系统的问题类别说明

类别	说明
其他	不可解决类别，如：这是什么？
介绍	内容型，如：物品历史、含义、景点介绍
人物	询问人物的，如：谁设计了天安门？
位置	位置类，如：天安门在哪里？地铁怎么去？
时间	时间类，如：天安门几点开门？
天气	天气类，如：北京天气如何？
价格	价格类，如：天安门门票多少钱一张？
尺寸	尺寸类，包括高度尺寸数量大小，如：天安门多高？
周边	周边相关，包括吃喝玩乐住，如：天安门附近宾馆？

致谢

时光荏苒，短短两年半的研究生生活已经接近尾声。这段时间也是我学生生涯的最后的两年半，它磨练了我的棱角，让我认识到了更为广大的学术研究世界，深深地影响了我的人生观、价值观。在本论文完成之际，我要真诚地向所有帮助过我的老师、同学、亲友表示感谢，正是你们的陪伴和鼓励，给我前进的勇气和动力，让我克服种种困难。

首先我要感谢我的导师王小捷教授。从本科开始，王老师的治学态度就深深地影响着我，使我对自然语言处理产生了浓厚的兴趣，并在大三末期毅然报考了王老师的研究生。研究生期间的学习生活虽然比较紧张，但是王老师仍为我们创造了宽松、自由、民主的研究环境。在学术上，王老师每周组织例会学习，在会上大家不单可以在积极的讨论中加深对自身领域的了解，还能了解其他同学的研究内容，大大扩宽我们的知识面。除了每周的例会，王老师还鼓励我们参加相关的学术比赛、学术报告和会议。在生活上，王老师担心我们忽略了身体的锻炼，经常组织集合体育活动，为我们预定羽毛球场地，丰富我们的大学生活。在此，再次对王老师对我平日的关心和指导表示感谢。

其次我要由衷地感谢袁彩霞、鲁鹏、李睿凡、李蕾等老师。在研究生期间，他们对我的学术上进行了深入地指导，令我感叹知识的全面性的重要性，并督促我加深对知识体系的理解。

然后我要感谢刘广、毛宇兆、汪悦、黎航宇、邝智杰、周雪、臧虎和钟可立学长学姐们，在研究生生活中，你们鼓励我挑战困难，指导我解决专业问题，并在求职过程中提供了许多帮助。在此向你们表示诚挚的谢意。

我还要感谢吴国华、梁忠平、冷冰、熊峰、高明辉、李东亮、罗瑾文和曾祯等同学以及我的舍友李茂林、张云飞、汪雷，正是你们的陪伴，让我度过快乐而美好的几年时光，使我的研究生生活丰富多彩。

特别地，我要感谢我的女友高芷乔，感谢你的包容和陪伴，让我感受到幸福的滋味，为我的研究生生活增加许多甜蜜的乐趣。

最后，我要感谢我的家人。感谢养育我多年的父母，正是你们从小对我的教育，让我充满信心，无惧风雨。感谢我的姐姐，感谢你多年对我的包容，小外甥女的出生让我感受到生命的奇迹，祝你成为最快乐的孩子。

作者攻读学位期间发表的学术论文目录

[1] Wu G, He D, Zhong K, et al. Leveraging Rich Linguistic Features for Cross-domain Chinese Segmentation[J]. CLP 2014, 2014: 101.

[2] Tan Y, X Yang, D He. Recognizing Textual Entailment Using Multiple Features[J]. Journal of Information & Computational Science, 2014, 11(1):181-187.

[3] Zhiqiao Gao, Lei Li, Liyuan Mao, Dezhu He. Content linking for UGC based on Word Embedding model[C]// Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2015 11th International Conference on. IEEE, 2015.