# 博士学位论文

# 中文词法句法语义联合模型研究

# JOINT MODELS FOR CHINESE MORPHOLOGICAL SYNTACTIC AND SEMANTIC PARSING

张 梅 山

哈尔滨工业大学

**2014年7月**

国内图书分类号：TP391.2　　学校代码：10213
国际图书分类号：681.324　　密级：公开

工学博士学位论文

# 中文词法句法语义联合模型研究

博士研究生：张梅山

导　　师：刘　挺　教授

申请学位：工学博士

学　　科：计算机科学与技术

所在单位：计算机科学与技术学院

答辩日期：2014年7月

授予学位单位：哈尔滨工业大学

Classified Index: TP391.2
U.D.C: 681.324

Dissertation for the Doctoral Degree in Engineering

# JOINT MODELS FOR CHINESE MORPHOLOGICAL SYNTACTIC AND SEMANTIC PARSING

**Candidate:** Meishan Zhang
**Supervisor:** Professor Ting Liu
**Academic Degree Applied for:** Doctor of Engineering
**Specialty:** Computer Science and Technology
**Affiliation:** School of Computer Science and Technology
**Date of Defence:** July, 2014
**Degree-Conferring-Institution:** Harbin Institute of Technology

# 摘 要

中文句子级别的分析技术是自然语言处理相关应用最基础的分析技术，它主要由词法分析、句法分析和语义分析三部分组成。其中词法分析包含分词和词性标注两个级联的任务；句法分析有短语结构和依存结构句法分析两种主流的分析手段；而对于语义分析，我们着重于语义依存分析。这些任务之间是存在着层次关系的，对于一个句子我们首先要进行分词，然后是词性标注，紧接着是短语结构或者依存结构句法分析，最后是语义依存分析。

传统的句子分析方法一般按照层次顺序依次使用各个任务最先进的模型进行处理，这种方法被称为串行的方法。它存在两个方面的问题：第一点是错误蔓延问题，即低层的错误会进一步扩散到高层；第二点是每层局部优化，因此低层的任务无法充使用高层的信息。这两个问题使得联合模型的方法得到了广泛的关注，它将多个层级相邻的任务放在一个统一的模型中来处理，从而避免这两个问题，因此能够提高各个任务的分析性能；同时它还可以使得自然语言处理的研究人员能更好的理解各个任务之间的相互关系。本论文中，我们对联合模型展开了四个方面的研究工作，分别如下所示：

1. 提出了一种基于字的中文句法分析方法，从而有效的将词法和句法分析结合在一起。根据中文的特点，我们发现大部分的中文词语存在着内部结构，而这一内部结构很少被前人利用。本文在基于这一词内部结构的基础上，将传统的基于词的句法树进行了扩展，得到了基于字的句法树。同时我们也对原有基于词的句法分析的算法进行了改进，得到了基于字的句法分析模型。我们分别从短语结构句法分析和依存结构句法分析出发，来实现了这中词法和句法的联合模型。最终的实验表明，这种基于字的句法分析模型能取得目前最好的性能，而且这种基于字的句法分析模型也能自动解析词的内部结构，在假定词被正确分析的情况下，词内部结构的准确率也达到了90%以上。

2. 提出了一个中文句法依存和语义依存的联合模型。中文句子级的语义分析方法一直是一个争议性很大的问题，本文采用了一种简单易用的依存文法来进行语义分析。首先我们介绍语义依存分析的定义以及相关规范，并且通过另一种被广泛使用的浅层语义分析手段——语义角色标注——来表明语义依存分析的有效性。然后我们提出了一种语义依存和句法依存分析的联合模型。在实验部分我们依次表明了语义依存分析的特点以及其有效性、然后验证了我们提出的语义依存和句法依存联合模型能取得更好的分析效果。

3. 提出了一种高效率高性能的中文词性标注依存句法的联合模型。目前主流的词性标注和依存句法联合模型有三种：基于图的联合模型、基于转移的联合模型和基于短语结构的联合模型，其中效率最高的方法也只能达到平均每秒9句的性能。我们通过模型融合和过训练相结合的方法来达到高效率高性能这一目的。一方面我们采用了基于栈学习的方法将这三种联合模型融合在一起，使得联合模型的性能得到了大幅度提升；另一方面我们采用了过训练的方法，通过融合联合模型自动分析大量未标注语料，并把这些自动分析的结果加入到一个简单快速联合模型的训练语料中，使得这一简单快速联合模型的性能大大增强。最后的实验表明，我们最终的联合模型能取得120句每秒的分析速度，而且其分析性能和最初的三种基本联合模型的水平相当。

4. 提出了一种基于词典和句子标注相结合的方法来提升分词词性标注联合模型的领域自适应能力。实验表明，在新闻领域下训练得到的最好的分词词性标注联合模型应用在文学小说领域时，性能会下降10%以上。语料标注是最有效的解决这一问题的方法，但是如何标注语料会最省劳动代价呢。本文探讨了两种常用的语料标注方法：词典标注和句子标注，它们都能有效的提升分词词性标注联合模型的领域自适应能力。这两种方法互有利弊，前人的实验表明在一些特定的环境下词典标注能取得更好的性能，这些比较都是建立在单独的词典标注和句子标注基础上的。本文提出将这两种方法相结合，即同时标注少量的词典和句子。实验结果表明，这种结合的方法在指定的标注代价下，能取得最好的领域自适应效果。

总体上，本文实际上对与联合模型密切相关的三个问题进行了展开工作，而且这三个问题是呈递进关系的。第一个问题是联合模型的建模方法问题，它也是联合模型最基本的问题，本文前两个方面的研究内容是针对这一问题而进行展开的。其次，虽然联合模型能够提升各个任务的性能，但是由于其解码复杂度的增加而导致了效率的严重下降，这使得联合模型效率和性能之间的平衡也是非常值得研究的问题，本文第三个方面的研究工作便是以词性标注和依存句法的联合模型为例来探讨这一问题的。最后，本文第四个方面的研究工作回到了自然语言处理领域的一个比较基本的问题：领域迁移问题，这一问题具有很大的挑战，而且一直也没有得到很好的解决，因此我们以最简单的联合模型——分词词性标注的联合模型——为例，探讨了这一问题。

**关键词：** 分词；词性标注；基于字的中文句法分析；中文语义依存分析；联合模型；

# Abstract

Sentence-level Chinese language processing is a fundamental module for applications of natural language processing (NLP). It includes Chinese morphological parsing, syntactic parsing and semantic parsing, where morphological analysis consists of word segmentation and POS-tagging, syntactic parsing has two major tasks: constituent parsing and dependency parsing, and semantic parsing refers to Chinese semantic dependency parsing in this paper. There is a hierarchy structure between these tasks. For a given Chines sentence, we usually conduct word segmentation first, and then POS-tagging, and thirdly constituent parsing or syntactic dependency parsing, and finally semantic dependency parsing.

Traditional methods process the above tasks by their state-of-the-art models independently. We usually call these methods by pipeline methods. They have two major drawbacks. First, they suffer the error propagation problem, where the errors in lower-layer tasks will spread to higher-layer tasks. Second, since they optimize a single model locally, lower-layer tasks can not use the information from higher-layer tasks. Because of the two problems, many researchers pay more attention to joint models, which process multiple adjacent tasks with a single model, so that the above problems can be avoided and improved performances can be achieved. Another advantage is that the joint models can facilitate language researchers to understand the relations between different tasks. In this paper, we study the joint models based on four points, as shown in the following:

1. We propose character-level models for Chinese syntactic parsing, so that the morphological and syntactic parsing can be processed jointly. For Chinese, majority words can have internal structures, which have been largely neglected in past works. Based on the internal structures of words, we extend the word-based syntax trees into character-level trees. And also we extend word-based syntactic parsing algorithms, so that the character-level trees can be handled. We consider both the constituent and dependency syntactic parsing. Final results show that our character-level parsing can achieve best performances. Specially, our character-level parsing models can analyze the internal structures of Chinese words. Our model can parse the internal words structures with an accuracy over 90% assuming words are correctly identified .

2. We propose a joint model for syntactic and semantic dependency parsing. The representation of sentence-level semantics is a controversial problem. We suggest the semantic dependency parsing, because it is simple and easy to use. First we describe the Chinese semantic dependency parsing and related specifications, and then we try to validate it by comparisons with another popular semantic representation method, semantic role labeling (SRL). Second, we propose a joint model for semantic and syntactic dependency parsing. In the experiments, we demonstrate the characteristics and reasonableness of the Chinese semantic parsing, and then show that our joint model can achieve better performances for Chinese syntactic and semantic dependency parsing.

3. We propose a high-efficiency and high-performance joint model for Chinese POS-tagging and syntactic dependency parsing. The mainstream models for joint POS-tagging and dependency parsing are graph-based, transition-based and constituent-based models, where the model with the highest efficiency can only achieve a speed of below 10 sentences per second. We use model integration and uptraining together to achieve our purpose. On the one hand, we exploit stacked learning to integrate the three models. obtaining a very high performance joint model. On the other hand, we use uptraining to add a large scale of auto-parsed sentences by the integrated model into the training data set of a simple and fast joint model. Final results show that our final joint model can achieve a speed of 120 sentences per second, with little decreases in performances compared with the baseline models.

4. We propose a combined type- and token-annotation to improve the domain adaption capability a joint word segmentation and POS-tagging model. Experimental results show that the tagging accuracy of a joint word segmentation and POS-tagging model well trained on news domain corpus has decreases of over 10% when applied to a literature domain. Corpus annotation is the most effective method for the problem. But how to annotate a corpus for domain adaption with the most efficiency? We investigate two annotation strategies: type-annotation and token-annotation. Both are effective ways for the problem. The two methods have advantages and disadvantages in different ways. Previous strategies show that type-annotation can be better in some special conditions. In this paper we suggest a combination of the two strategies. Final results show that this combination can achieved best performances under a fixed cost.

In conclusion, this dissertation actually considers three problems that have close relations with joint models. These problems are gradually higher-level for joint models.

First we consider the modeling methods, a most basic problem. The first two works of our paper aim for this problem, which implement the joint morphological-syntactic and the joint syntactic-semantic models, respectively. Second since the joint models are more complex than the pipeline models, which make the decoding speed much slower. Thus the balance between the performance and the efficiency is an important problem. We study the problem by the third work of this paper, taking joint POS-tagging and dependency parsing as an example. Finally, the fourth work of our paper aims to domain adaptation, which is a popular problem in NLP. The problem is challenging and still unresolved. Thus we take the simplest joint model (joint word segmentation and POS-tagging) as an example to study the problem.

**Keywords:** Word Segmentation; POS-Tagging; Character-Level Chinese Parsing; Chinese Semantic Dependency Parsing; Joint Models

# 目 录

# Contents

# 第 1 章 绪论

## 1.1 课题背景及意义

### 1.1.1 课题背景

中文自然语言处理句子级别的基本分析技术，主要包括词法分析、句法分析、语义分析三个大方面。其中词法分析包含分词和词性标注两个子任务；句法分析一般分为两类，短语结构句法分析和依存结构句法分析；语义分析在本论文主要针对语义依存分析。本文的主要研究目的是分析这一系列基本技术，并通过联合模型的手段使得它们的性能或者效率得到更进一步的提升。

目前这一系列基础分析技术主要采用逐层递进处理的方法，包括四个步骤，如图1-1所示。首先第一步为分词，然后是词性标注，紧接着是短语结构句法分析或者依存结构句法分析；最后是语义依存分析。这些基本的分析技术，是对中文句子分析逐步加深的过程。

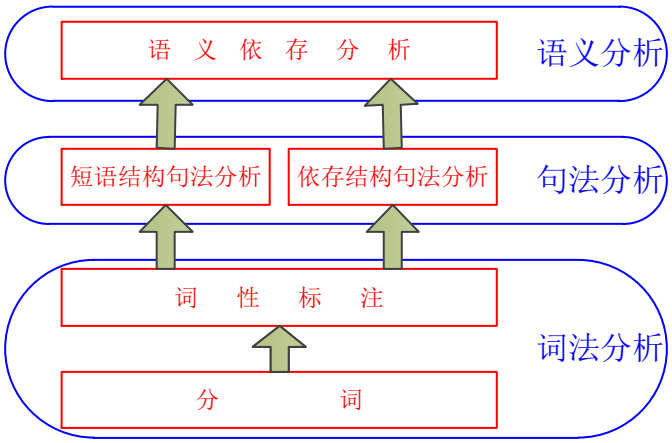



图 1-1 中文自然语言处理句子级别的分析技术层级结构示意图。

Fig. 1-1 The hierarchical structure of the sentence-level Chinese language processing.

这种逐层递进的串行处理方法面临着两个主要的问题，使得最终的句子分析性能无法达到最佳。首先，一旦低层任务的分析结果出错了，高层任务的分析结果则不可避免的受到影响，这就是错误蔓延问题。其次，很多情况下低层任务自己很难判断它目前的分析结果是否合理，但是有了高层任务的信息，这个判断就会变得容易很多，但是在串行模型中我们很难利用到这种高层的信息，因为它采用了逐层局部优化的分析方式。

鉴于上面的两个问题，本论文通过使用联合建模的方法来加以解决。对于两个任务，其中一个任务是另外一个任务的基本输入的情况，我们对这两个任务使用一个统一的模型进行联合分析，使得两个任务能互相充分的利用，进而它们的性能都能得到提升，我们将这样的模型称为联合模型。这种联合模型的方法已逐渐受到了相关研究者的重视，例如Wenbing Jiang 等人以及张岳等人提出的分词词性标注联合模型[1–3]，李正华等人提出的词性标注与依存句法分析联合模型[4]等等。

联合模型不仅对提升句子分析的性能有着积极的作用，还能使得研究语言的学者能更好的理解中文词法、句法以及语义各部分之间的相互关系，例如分词与词性的关系，词性与句法的关系以及句法与语义的关系等等。

### 1.1.2 课题意义

中文句子级别的各项基础分析工作，对自然语言处理的相关应用都能够提供各种直接的或者间接的帮助，这里分别从词法分析、句法分析以及语义分析三方面逐项进行说明。

(1) 词法分析

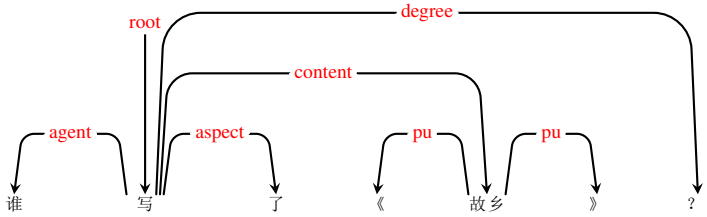
词法分析包括两步，分词和词性标注。词法分析是中文自然语言处理的基础，中文处理中的各项关键技术都离不开词法分析，例如句法分析以及语义分析。另一方面，对于自然语言的相关应用，词法分析也是非常重要的一步。例如信息检索中的查询字段(Query)分析，词法分析的结果会一定程度上影响分析的效果，如果“发展/中/国家”，被错误切成“发展/中国/家”，则势必会影响最终检索的性能。

(2) 句法分析

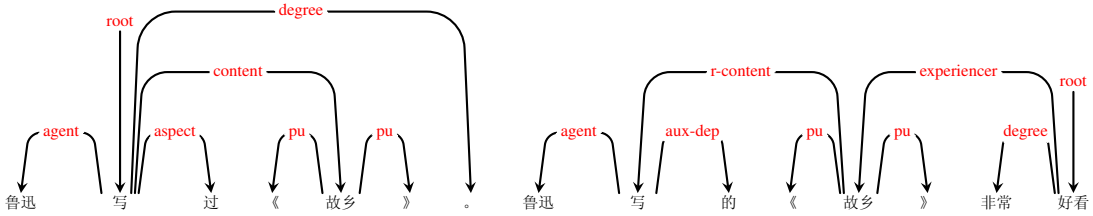
句法分析主要包括短语结构句法分析和依存结构句法分析，两者在一定程度上可以相互转换。在机器翻译、自动问答、信息抽取等应用中，已经有了不少工作将句法分析的结果作为特征来提升系统的性能。例如基于短语的机器翻译，主要依据短语对齐的结果，而准确高效的句法分析可以提高短语对齐的准确率，从而改善机器翻译的效果。在基于自然语言的自动问答中，查询扩展以及答案匹配均需要对句子进行深入的理解和分析。一些工作将依存分析用于自动问答的问题分类中，取得了较好的效果，表明了句法分析对自动问答的重要作用。句法分析的另一个直接应用是信息抽取，为了从非结构化的文本中自动抽取特定的结构化信息，句法分析的作用至关重要。

(3) 语义分析

语义分析能比句法分析更进一步的支持类似于自动问答、信息抽取这样的应用，这种帮助是非常直接的。我们以语义依存分析为例来说明语义分析对自动问答这种应用的直接帮助作用。例如对于问题“《故乡》的作者是谁?”，答案为“鲁迅”。如果使用语义依存分析技术，识别出动词的施事、受事以及该动作发生的时间、地点等信息，这对回答上面的问题是非常直接的。图1-2展示了语义依存分析在自动问答系统中的应用。我们对用户问句进行语义依存分析，结果如图1-2 a)所示，得知问题所要找的答案是谓词“写”的施事，限制条件是“写”的内容是“故乡”。通过语义依存结构以及关系的匹配，我们能从图1-2 b)和图1-2 c)中均能得到前面问题的答案是“鲁迅”。




a) 用户问题




b) 句子1　　c) 句子2

图 1-2 语义依存分析在问答上的应用。

Fig. 1-2 Question answer using semantic dependency parsing.

最后值得注意的是联合模型的方法也能使得语言学家加深对中文处理各个步骤之间相互关系的理解，从而从中得到一些提示，进而进一步改善语言分析的方法，例如对分词、词性标注、句法分析以及语义分析的标注规范以及表示方法的改善。

## 1.2 研究现状及分析

中文自然语言处理句子级别的基础分析技术主要包括三部分：词法分析、

句法分析和语义分析。其中词法分析包括分词和词性标注两个任务。在最近的十年里，基于统计机器学习的方法在这些分析任务中占据了主流的地位，其主要原因在于这类方法使得这些任务的性能远远超过了以前基于规则的启发式方法。在句子级别的中文词法句法以及语义分析中，主要的统计模型方法可以分为两种，基于图模型的分析方法和基于转移的分析方法。基于图模型的算法，主要是将句子分析任务看成一个图解码问题，使得句子分析的过程被转换为从一个带权重的图中搜索满足特定约束的分数最高的子图。这个解码过程和具体图的表示方法以及相应的马尔可夫假设有关（马尔可夫假设使得我们能使用动态规划的算法对任务求解）。对于基于图的模型，不同的任务存在着不同的建模方法，我们将针对各个任务对这一类方法的国内外研究工作的进行描述。

基于转移的算法，是将句子分析看成一个有限状态自动机从初始状态到达最终状态的动作转移过程。任何一个状态在面临一个转移动作时都会有一个具体的打分函数为该动作进行打分，在统计模型中，这一打分函数由这一动作在该状态下产生的特征所决定，这样最终将句子分析转换成搜索一组累加分数最高的从开始状态到达最终状态的动作转移序列。对于基于转移的统计分析模型，总体框架相对于基于图的统计模型来说比较固定。首先具体搜索最优解可以采用贪心算法或者柱搜索(Beam search)算法，其中贪心算法总是找到当前状态下分数最高的下一个动作，从而得到下一个状态，然后依次下去，得到最终的分析结果。而柱搜索算法每一步都是根据当前的状态候选集合产生分数最高的固定数目的下一步转移状态候选集合，前面的基于贪心的算法只能找到局部最优解，而柱搜索算法能找到全局次优解，而且贪心算法是柱搜索算法的一种特殊情况，即柱大小为1的情况，因此柱搜索算法能取得更好的性能。

然后对于基于转移的算法中模型参数的训练方法，在采用贪心搜索时，一般采用分类算法，即输入为一个转移状态，而输出类别为一个转移动作，具体的分类算法可以采用最大熵分类算法或者支持向量机(Support Vector Machine,SVM)算法[5, 6]。而对于柱搜索算法，一般都采用在线的感知器算法结合提前更新[7, 8]，在解码的过程中，如果发现正确答案在没有办法由柱中的任何一个状态生成时，则对特征权重进行更新，它和一般的在线感知器算法等到解码完毕之后才进行更新是有着一定的区别的。

将这种基于转移的算法应用在句子级别的中文词法句法和语义分析时，我们只需要根据具体的分析任务定义其转移状态以及一个转移状态所能面临的操作的集合即可,定义方法视具体任务而定，我们将在各个任务的国内外研究工作中分别描述它们的定义方法。

这一节的安排包含四部分，首先我们对中文词法、句法以及语义分析各自的任务以及相应的国内外研究现状进行详细的介绍，然后在第四部分我们进一步阐述了这些任务上联合模型的国内外研究工作。

### 1.2.1 词法分析

词法分析包括分词和词性标注两部分，本节分别对这两种任务的相关研究进行描述。

**1. 分词**

中文和英语等语言有着很大的一个区别，即在中文的原始句子中，词与词之间不存在明确的界限，因此在开始中文句子分析时，分词是一个首要解决的问题。分词就是把中文的原始句子分割成词序列的过程。例如对于句子“新加坡印尼已签订航空协定”，经过分词之后，我们得到“新加坡 印尼 已 签订 航空 协定”。目前主流的基于统计的分词方法可以细分为三类。

(1) 基于字的模型。

基于字的分词方法将句子中的每一个字采用分类的方式进行处理，例如句子中的每一个字都被区分为BMES 四种标记之一，其中B 表示该字符位于一个词的开始位置，M 表示该字符位于一个词的中间位置，E 表示该字符位于一个词的结束位置，而S表示该字符本身是一个单字词，例如“新加坡 印尼 已 签订 航空 协定”这一分词结果对应的的序列标签为“BMEBESBEBEBE”。该方法最早由薛念文等人提出，他们使用了最大熵分类模型来进行字符分类[9]。后期更多的研究者尝使用更复杂的序列标注模型来对上面这一问题进行求解，这是一种基于图的统计模型。这一类算法建立在一元马尔可夫基础之上，即假设目前时刻的标签只与前一时刻的标签相关，这样在解码时便可以使用基于动态规划的Viterbi算法。这一解码算法是针对所有序列标注任务的，其中特征的参数学习方法包括条件概率随机场(Conditional Random Field，CRF), 感知器算法(Perceptron)等等[7, 10, 11]。

(2) 基于词的模型。

前面所提到的方法很难将词相关的特征融入到分词中去，而直觉上词应该对分词有很大的帮助，最早期的规则系统实际上就是一种基于词的方法。在统计模型中，加入词特征会导致解码的难度大大增加，因为一个句子中可能的词的个数是随着句子中字的数目呈指数级别增长的。最初部分研究者提出使用词类和语言模型相结合来预测某个分词结果的概率，例如高剑锋等人在计算语

言学(Computational Linguistics, CL)2005年的工作[12]。

另一种解决句子中词指数级搜索空间的方法是采用基于转移的系统，这个系统最早由张岳等人在2007年提出[13]，其中转移状态由一个栈和一个队列组成，栈中存储着已经部分解码的词序列，而队列中存储着尚未处理的字序列。转移动作共有两类，一种是分开(Separate)，即将下一个字符移入栈中作为一个单独的词，在使用这一动作时，我们可以非常方便的加入各种词相关的特征；另外一种是附加(Append)，即将下一个字符移入栈中并且和栈顶的词合并。

(3) 字词混合模型。

这类模型一般使用模型融合的方法进行实现，相关的工作包括Ruiqiang Zhang等人提出的基于子词(Subword)的分词方法[14] 和孙薇薇提出的基于字的与基于词的融合模型[15]。

**2. 词性标注**

词性标注是为中文句子中的每一个词指定其类别的一个任务，词性反映了一个词的类别，主要的词性类别包括名词、动词、介词、标点、形容词、副词等等。词性标注问题是一个典型的序列标注问题，比较早的词性标注分析器采用了隐马尔可夫模型[16]，它是一种生成模型，一般建立在二元文法的基础之上，主要估计两种不同的概率，词性与词性之间的转移概率以及词性到词之间的发射概率，这种模型的性能受限于特征的使用，因此部分研究者尝试使用潜变量的方法对这一模型作出改进[17]。

由于生成模型只能使用非常有限的特征，因此很多人都采用了判别模型的方法来进行词性标注，这类方法在解码时假设词性序列满足二元马尔可夫性，这样便能使用前面提出的Viterbi算法进行解码，然后使用最大熵马尔可夫算法、CRF模型或者在线感知器算法等等[18–20]，进行参数学习。

词性标注也可以由基于转移的算法来处理[21]。在词性标注的转移系统中，其转移状态包含一个栈和一个队列，栈中存储着已经标注了词性的词序列，而队列中是尚未处理的词序列；其转移动作带参数的移进操作，称之为SHIFT($t$)，该动作每次将队列中的一个词语移入栈中，并赋予该词的词性为$t$。在这样的转移系统中，并不需要词性转移的二元马尔可夫假设，从而可以融入更多的特征，包括词性的三元组特征或者更高元组特征。

### 1.2.2 句法分析

目前主流的句法分析分为两种，短语句法分析和依存句法分析，其中短语

句法分析采用上下文无关文法为句子的句法结构建模，而依存句法分析则采用一种基于词的依存文法来对句子的句法结构进行建模，本小节将分别对这两个任务以及相关研究工作做简要总结。

**1. 短语句法**

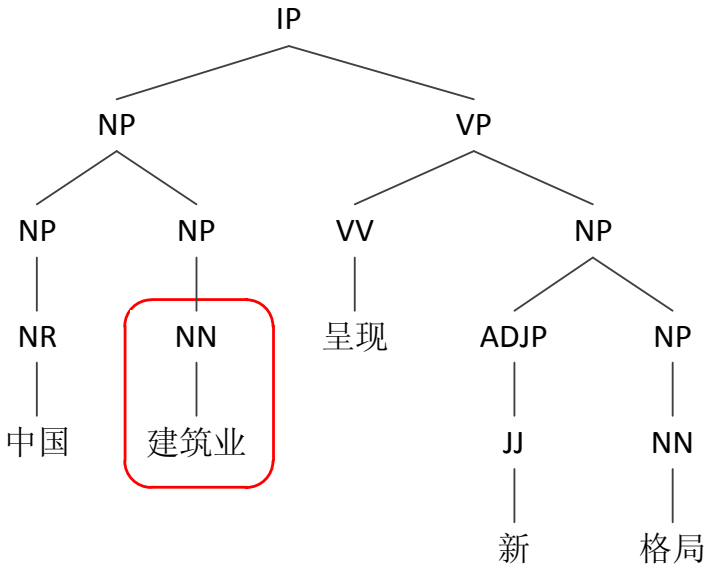



图 1-3 短语句法分析器输出示例。

Fig. 1-3 An example of Chinese constituent parsing.

对于给定的句子，短语结构句法分析的结果如图1-3 所示。短语结构句法分析建立在上下文无关文法的基础上，上下文无关文法可以定义为四元组$< W, G, \text{ROOT}, \text{RULE} >$，其中$W$表示终结符的集合，也就是词的集合；$G$表示非终结符的集合，也就是成分标记以及词性的集合；ROOT 表示充当短语句法树根节点的标记，而RULE表示文法规则的集合，其中每条文法规则可以表示为$A \rightarrow A_1 \cdots A_n$，这里$A$为非终结符，$A_i$属于终结符和非终结符的集合。一般在实际的短语句法分析系统中，为了提高分析的效率，都会采用一定的方法对文法规则进行二叉化，使得最终的短语句法树中只包含一元产生式和二元产生式(即$n \leq 2$)，而且转化方式必须是可逆的，即我们可以将转化后的短语结构句法树通过一些特殊标记还原成原来的树。

短语句法结构分析模型可以划分为词汇化的和非词汇化的分析方法。词汇化的短语结构句法分析为上下文无关文法引入了词汇信息，来对产生式进行细化，例如对于产生式$A \rightarrow A_1 \cdots A_n$，经过词汇化之后，该产生式转化为$A \circ w \rightarrow A_1 \circ w_1 \cdots A_n \circ w_n$。该方法最主要的突破来自于Collins 句法分析器[22]，它使得短语句法分析的性能得到了很大的提升。进一步，在Collins

句法分析器的基础上，Charniak句法分析器以及二阶段重排序(Reranking)算法使得句法分析器的性能达到了最好的效果[23]，其解码算法采用基于图的Cocke–Younger–Kasami（CKY）算法。

词汇化的短语结构句法分析也可以通过基于转移的系统来实现，最早的中文短语句法分析转移系统由王梦秋等人在2006年提出[24]。在基于转移的短语句法分析系统中，其状态由一个栈和一个队列组成，栈中存储着部分解码的短语句法树序列，队列中是尚未处理的词序列。在该系统中，一共定义了三类转移动作，移进（将队列中的词移入栈中），一元归约（为将栈顶的短语句法树添加一个一元的短语产生式）和二元归约（为将栈顶的两个短语句法树添加一个二元的短语产生式）。他们使用了贪心搜索的方法来进行短语句法分析，进一步，张岳等人将贪心算法改成柱搜索算法[25, 26]。

非词汇化的句法结构分析采用符号重标记的方法来进一步细分短语句法标签。最简单的方法是将短语结构树中任意父亲节点的标记挂载到非终结儿子节点上，以扩大上下文的范围。Klein和Manning在2003年，利用语言学的知识，使用人工的方法对短语树库中的非终结符号进行了细分，取得了一定的性能提升[27]。Matsuzaki等人首次提出了使用自动的方法[28]，对短语树库中的非终结符号进行细分类，这种方法固定将每个非终结符号分成8类，最终也取得了一定效果。这几种符号重标记的方法，和不进行重标记相比，取得比较好的效果，但是这些方法和基于词汇化的方法相比，性能上不具有优势。取得突破性进展的工作是Petrov和Klein等人采用自动切分-合并(Splitting-Merging)的方法对非终结符号进行重标记[29]，在具体如何切分-合并时，应用最大期望算法（Expectation–Maximization,EM），这样得到的句法分析器(Berkeley Parser)获得了优于词汇化方法的句法分析器。其后，Petrov等人进一步提出了基于隐含标记文法的由粗到精（Coarse-to-Fine）解码算法[30]，对上述算法进一步改进。由于应用最大期望算法时需要设置一组初始参数，而这组参数对最终模型的性能影响比较大，因此Petrov 等人再进一步进行了优化，采用多个初始值，每一个初始值对应一个句法分析模型，最后将这些模型进行融合。

**2. 依存句法**

依存句法分析是建立在依存语法的基础上，它描述的是句子中词与词之间直接的关系，这种关系称为依存关系（Dependency Relations，或者简称Dependencies）。一个依存关系连接两个词，它是存在方向的，分别被描述为核心词（head）和修饰词（dependent），核心词所对应的节点为父亲节点，而修饰词所对应的节点为孩子节点。利用依存语法进行自动语义分析得到的是一棵

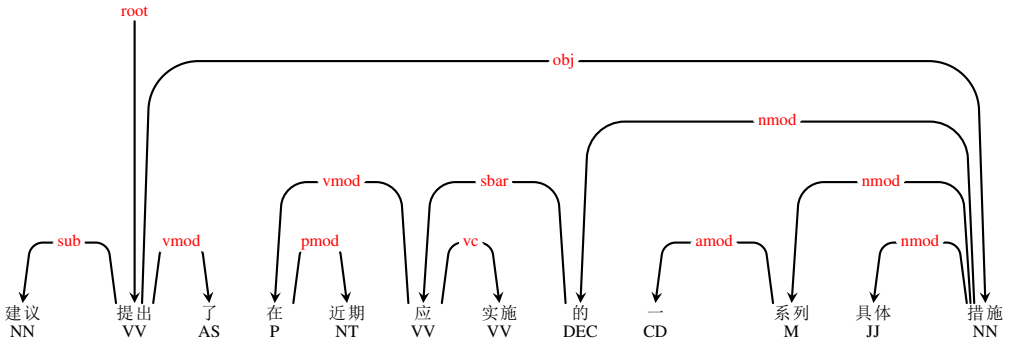



图 1-4 依存句法分析器输出结果示意图。

Fig. 1-4 An example of Chinese syntactic dependency parsing.

依存树，这棵树中不含非终结节点，只有由具体词构成的终结节点。对于给定的一个句子，依存句法分析器的输出结果如图1-4所示。一般来说，一棵中文依存句法分析树需要满足以下三个条件：(1)每棵树中有且仅有一个根节点，该节点的父亲节点不在句子内部；(2)根节点以外的任何词都有且仅有一个父亲节点；(3)将一棵依存树按词序平铺开，不存在任何交叉弧。

依存句法分析的目的是对任意给定的一个句子，解析得到其依存句法树，其方法主要可以分为以下三类：

(1) 基于图的算法。

这种方法，将依存句法分析看成从一个有向多重图中寻找分值最大的那棵依存句法树。这种方法假定句子对应的依存树中，只有某些子树（subtree）中的依存弧之间才具有互相联系和影响，而与其他依存弧之间则互相独立。基于这种假设，一棵依存树的分值便可以分解成若干子树分值之和，从而我们便能使用动态规划的算法进行解码。根据子树中弧的数目的多少，解码算法可以分为一阶、二阶、三阶或者高阶算法[31–34]，随着阶数的增加，算法复杂度也逐渐提升。在模型参数训练时，我们一般使用在线的平均感知器算法。

(2) 基于转移的算法。

在基于转移的依存句法分析系统中，和分词、词性标注以及短语句法分析类似，其状态都是由一个栈和一个队列组成，只不过栈中存储着部分解码的依存句法树序列，队列中存储着尚未被处理的词语。对于转移动作，有两类不同的算法，一种是标准弧转移算法，另一种是贪心弧转移算法。对于前者，定义了四种转移操作，分别为移进、左弧、右弧和根出栈，移进是将队列中的词移入栈中，左弧和右弧是将栈顶的两个依存树进行合并，使得其中的一棵树成为

另一棵树的孩子，根出栈是当栈中只剩一棵依存树而且队列为空时，标记依存分析完成的一个操作。对于后者，定义了五种操作，分别为移进，左弧、右弧、根出栈和归约，前四种和标准弧转移算法类似，只不过左弧和右弧操作发生在栈顶和队列头部，归约是将栈顶的元素移出栈，标志这个元素所有的孩子以及父亲节点都已经找到。

最初，大部分研究者都采用基于分类的贪心算法去逐步预测符合当前上下文的最优动作，例如使用最大熵模型，支持向量机模型等等[35–37]，这种贪心的方式和基于图的算法在性能上还有一定的差距。进一步，黄亮等人(2009)以及张岳和Clark(2008)分别将标准弧转移算法和贪心弧转移算法与柱搜索结合起来[25, 38–40]，大大提升了依存分析的性能，从而使得基于转移的算法在性能上和基于图的算法相当。

(3) 基于短语句法结构的算法。

这类算法分为三步，第一步是将依存句法根据规则自动转换成伪短语句法结构，这个转换过程必须是可逆的；第二步是使用短语句法分析器，例如Berkeley Parser，自动分析出伪短语句法结构来；最后一步是将伪短语结构转换成依存结构。孙薇薇等人在2012 年提出了两种依存结构到短语结构的转换方法[41, 42]，一种是直接利用中文宾州树库(Chinese Treebank, CTB)中的短语句法结构，另一种是基于自动的规则编码将依存树转换成一种短语结构句法树，这两种转换方法都取得了与基于图以及基于转移的系统相当的性能。

### 1.2.3 语义分析

语义分析是中文句子级别分析的终极目标。对于语义分析，不同的研究者有着不同的观点，这些观点主要分为两种，其中第一种是词汇语义的观点，基于词的层面来解析某个词在句子中的含义，词的词义解释一般建立在一个知识库上，例如英文中的Wordnet和中文中的Hownet以及同义词林相关的工作[43–45]，和词汇语义相关的自然语言处理任务主要是词义消歧[46, 47]；第二种是逻辑组合语义的观点，这种语义分析方式尝试将一个句子里面的词进行逻辑组合，最后将一个句子转换成为一串逻辑表达式。一般现在使用最广泛的逻辑表达式为lambda calculus[48]，具有代表性的工作包括Collins 等人提出的使用CCG 作为中间句法，将一个句子自动转换成逻辑表达式[49]；还有Percy Liang等人提出的基于依存的组合逻辑语义[50]。

上述的两种语义分析方法都比较极端，对于词汇语义的观点，在进行句子

级语义分析时便是一种非常浅层的语义分析手段；而对于第二种，由于其复杂的目标逻辑表达式导致了语义标注的难度非常大，因此这种分析手段面临着严重的语料问题，现有的研究工作往往只能在少量的语料集上进行。在本论文的研究中，我们倾向于一种折衷的语义分析手段——语义依存分析。这种方案是一种浅层语义表示，是词汇语义和逻辑语义之间的一个平衡，它可以作为迈向深层逻辑组合语义的第一步。

实际上，近年来广泛使用的语义角色标注也可以看作逻辑语义和词汇语义的一个折中[51–54]。语义角色标注是为句子中的谓词去寻找论元参数，同时指定该论元的属性，其中的谓词一般是动词或者名词。谓词论元参数的确定能体现一个词语的词汇语义，而具体每个论元参数的值则体现了逻辑语义。这种方法和语义依存分析有着很大的区别，其中最主要的区别包括两方面，一方面语义依存分析语义标签的粒度要比语义角色标注多出很多，另一方面，语义依存分析是针对句子中的所有词进行分析，并不只是针对语义角色标注中的谓词。

中文语义依存分析，实际上就是借助于依存语法的形式，来表示中文句子的语义，图1-5显示了一个典型的例子。该表示方法把句子用一个依存树表示，一定程度上包含了词与词之间的语义组合关系，同时每个具体的词所表达的语义也在树中体现了出来，因此这种方式既能一定程度上体现组合语义的观点，又能一定程度上表现词汇语义。

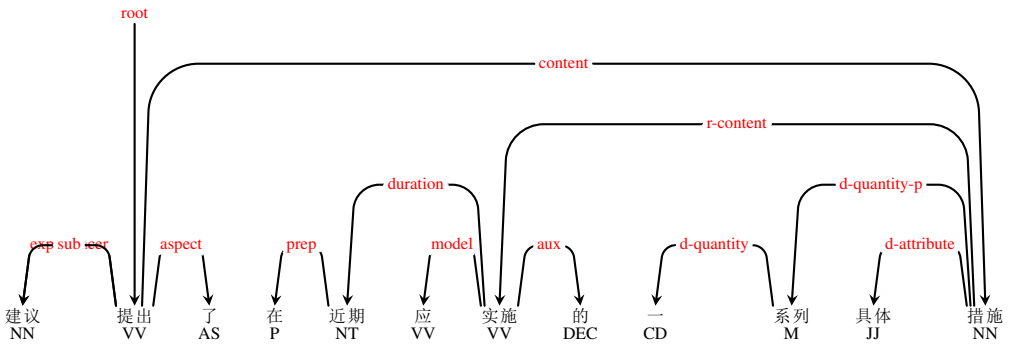



图 1-5 语义依存分析的一个例子。

Fig. 1-5 An example for Chinese semantic dependency parsing.

中文语义依存分析的相关工作相对比较少，主要都集中在语料库建设上面，其中所使用的语义标签大部分都是根据知网中的标签定义结合其它语义规范而形成的[44]。首先是李明琴等人2003年人工标注了100万词规模的语义依存语料[55]，然后Jiajun Yan在2007年手工标注了来自中文宾州树库中的4,000个短句

子（含有32000个词）[56]，最近车万翔等人也发布了他们所标注的基于中文宾州树库的10,086个句子的语义依存标注，并在此基础上组织了公开的语义依存评测[57]。

对于语义依存分析的算法，由于其表示形式和句法依存分析没有任何区别，因此任何句法依存分析的算法，都可以直接应用在语义依存分析上面，包括基于图的算法，基于转移的算法和基于短语句法结构的算法。值得注意的是，虽然语义依存分析和句法依存分析在算法上面完全一样，但是这两者具有本质区别的，语义依存分析语料的构建是直接面向语义的，考虑了中文的意合特性。

### 1.2.4 联合模型

前面介绍的是词法、句法以及语义分析各自分析技术的前沿工作，接下来的内容中，我们将对已有联合模型的前沿工作作简要总结和介绍。

**1. 分词和词性**

分词词性标注联合模型的建模方式可以分为两种，一种是采用基于序列标注的图模型进行扩展，另一种是采用基于转移的系统进行扩展。基于序列标注扩展的分词词性标注联合模型最早由Hwee Tou Ng和Jin Kiat Low在2004年提出[58]，由于分词和词性标注都可以看做序列标注问题，因此最直接的方法是将两种标记拼接在一起，形成一组新的输出标签，这种方法的缺点是时间复杂度特别高。进一步有人提出了重排序的方法[1, 59]作出了改善，也有人提出了基于子词预处理的方法[60]作出改善，这些方法不仅在速度上有了优势，而且准确率也进一步提高。

使用基于转移的系统来建立分词词性标注的联合模型最早由张岳和Clark等人在2008年提出[2]，随后他们在2010年将这一方法进行优化使得效率大大提升[3]。在基于转移的分词词性标注联合模型中，转移状态是一个栈和一个队列，栈中存储着已经部分解码的词和词性序列，队列中存储着尚未进行处理的字序列，其转移动作和分词的转移系统非常类似，唯一不同的是在分开操作时引入了一个词性参数，使得句子中的每个词在其开始字符在移入栈时被赋予了一个词性。

**2. 词性和句法**

我们从短语句法分析模型和依存句法分析模型进行分别介绍。短语句法分析和词性标注的联合模型工作很少，唯一例外的是类似于Berleley Parser短语分

析器的做法[30]，它直接将词性到词这一层看做一种特殊的短语文法来对待，因此它能同时分析出词性和短语语法的结果来。

最早的词性标注和依存句法分析的联合模型由李正华等人在2011年提出[4]，随后他们在2012年对这一模型作出了改进[61–63]。这种方法在基于图的依存句法分析上进行扩展，将每个词的词性也作为参数引入到解码算法中。这一方法存在严重的速度问题，他们采用了各种剪枝算法，最后得到的联合模型其速度能达到接近每秒1句。

对于依存句法分析和词性标注，同样也可以使用基于转移的方法来实现其联合模型，这一实现方式最早由Jun Hatori 等人在2011 年提出[64]，其后Bohnet等人在2012 年进一步对该算法提出了一些改进[65]。基于转移的依存句法和词性标注联合模型只能在标准弧转移算法上进行扩展而实现，其状态仍然由一个栈和一个队列组成，栈中存储着部分解码的依存树而队列中存储着尚未进行处理的词序列；其转移动作和基于转移的依存分析算法非常类似，只是在移进操作时，增加了一个参数，为移入的词赋予词性。

词性标注和依存句法的联合模型同样也可以使用短语句法分析和词性标注的联合模型来实现，只需先将依存句法结构按照孙薇薇等人提出的方式（这种方式是可逆的）先转化成为伪短语结构，然后在使用短语句法分析和词性标注的联合模型得到其短语句法树，最后再转化为依存句法结构[41, 42]。

**3. 分词词性和句法**

在分词、词性标注以及短语句法的联合模型方面，最早的系统由罗小强在2003年提出，他将一棵短语句法树进行改造，使得其叶子节点不再是词，而是字，即从词性到词时，增加了一层伪造的词性到字的树结构，增加的一层被标记为$l_1 \circ l_2$，其中$l_1 \in BMES$，$l_2$为词性，这样便形成一棵基于字的短语句法结构树[66]；李中国在2011 年基于他自己的标注也提出了一种分词词性和句法的联合模型。这两种方法都采用了生成模型，而且最终的结果都没有明显的改善[67]。进一步的，Xian Qian和Yang Liu 在2012年提出了一种联合解码的方法[68]，训练时还是使用三个各自的模型，只是在解码时，将分词、词性和短语句法结合在一起，实验的结果表明这种方法能有效的提升这三个任务的性能。

基于依存分析的分词、词性和句法分析联合模型也有一定的研究，Jun Hatori等人在2012年，将其基于转移的词性依存句法联合模型进行扩展，将分词任务也联合在其中，取得了不错的效果[69]。李中国和周国栋在2012年也用类似Jun Hatori的方法提出了将分词、词性和依存句法结合在一起，但是他们在分词词性上的性能没有得到明显的改善[70]。

**4. 句法和语义**

对于句法和语义联合模型的研究，前人工作大都是建立在句法分析和语义角色标注上面的[71]，但这一类工作很少能同时提升句法和语义分析的性能。

## 1.3 联合模型的挑战与前景

中文词法分析、句法分析以及语义分析各自的任务都有着自己的解码方式，如何把两种或者多种任务的解码算法结合在一起，得到一个非常有效的联合解码算法? 这一问题是建模方法方面问题，也是联合模型最基本的问题。另外多个任务的联合解码使得搜索空间变为各个子任务搜索空间的乘积，从而导致了解码效率的严重下降，这样如何提升联合模型的效率或者如何在联合模型的效率和性能之间作出一个平衡也是一个非常重要的问题。由于我们在研究中文自然语言处理句子级别的基础分析任务，其中一个熟知的难题，即领域迁移问题，是否在联合模型中变得更复杂了呢? 答案是很显然的，由于我们一下子处理多个任务，这样和以前的串行模型相比，联合模型的领域自适应由多了很多不确定性的因素。总体上来说，这三个方面都给联合模型带来许多挑战和前景，都是非常值得去探讨和研究的问题。

(1) **联合模型的建模** 目前中文词法句法和语义之间比较成熟的联合模型包括分词和词性标注的联合模型，词性和依存句法的联合模型，分词词性和依存句法的联合模型，这些联合模型都取得了一定的效果，在其与每个联合模型各个相关的任务上都有了一定性能的提升。但是仍然存在着很多任务，联合解码策略还可以得到进一步的改善，例如分词词性和句法的联合模型；同样也存在一些任务，由于解码算法的差别太大或者任务定义不完善而导致了联合模型实施难度非常大，比如句法和语义的联合模型。

(2) **联合模型效率和性能之间的平衡** 一般情况下，大部分研究者主要关注联合模型的性能，但在实际情况中，效率也是非常重要的一个因素。如果联合模型速度太慢，会导致一些对实时性要求比较高的应用无法使用这一成果。目前的一些实验表明，分词词性标注联合模型的速度要比普通串行方法慢5倍左右，而词性和依存句法的联合模型要比串行模型慢20倍左右，这些都将成为联合模型实际应用的一个瓶颈。如何能使得联合模型的分析效率有较大的提升但是仅允许有非常少量的性能下降呢? 这个方面的研究也至关重要。

(3) **领域迁移问题** 我们知道在自然语言处理中领域迁移问题是最难的一个问题，最有效的解决方法是标注少量的目标领域数据，对于联合模型来说由于

它面对了多个任务，所以标注难度和串行模型相比大幅度增加。那么我们能否有更好的适合于联合模型的领域迁移方案，或者是否存在一个更有效的领域语料标注方法呢?

## 1.4 本文的研究内容及章节安排

本论文尝试对中文词法、句法以及语义分析的若干相关任务进行联合模型的研究，涉及到的问题包括联合模型的建模方法、联合模型效率与性能之间的平衡以及联合模型的领域迁移问题。这三个问题是对联合模型相关研究高度逐步增加的过程，其中建模方法是最首要解决的问题，其次再考虑性能和效率的平衡，最后回到自然语言处理中一个一般化的问题——领域自适应能力。对于联合模型还不完善的相关任务，我们为其提出更好的联合模型建模方法；对于联合模型基本完善但是存在效率瓶颈的相关任务，我们为平衡其性能和效率提出解决方法；而对于联合模型基本完善的相关任务，我们基于相应的联合模型提出提高领域自适应能力的方法。在本论文中，我们对这三个问题做了四个方面的研究工作，其中前面两个方面的研究工作是从联合模型建模的角度来考虑，分别提出了词法与句法分析的联合模型以及句法和语义分析的联合模型。第三方面是从联合模型性能和效率平衡的角度进行出发，并且考虑到实际的效率变化大小之后，以词性标注和依存句法的联合模型为例（由于这一联合模型和串行模型相比，分析效率下降非常明显），展开了这一研究。第四个方面是从领域迁移的问题出发，考虑到领域迁移这个问题本省的难度，我们选择了一个最简单的联合模型——分词词性标注的联合模型——展开了研究。整篇论文结构框架如图1-6所示，具体的，本论文共分为五章，各章内容组织如下：

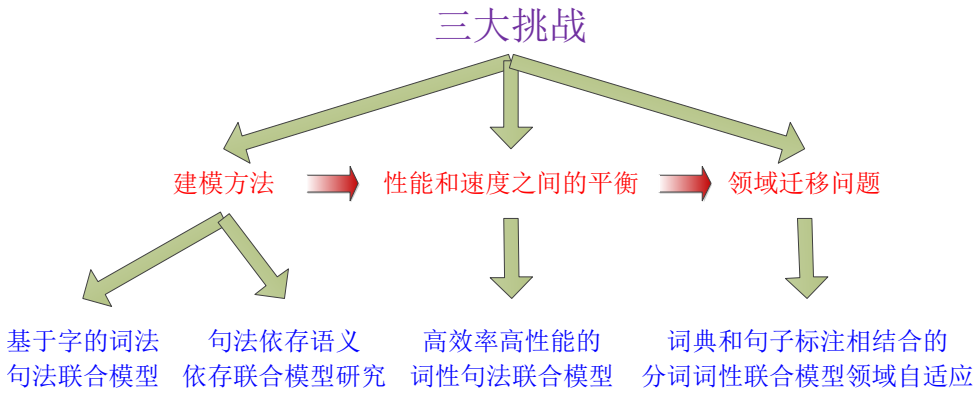



图 1-6 论文总体框架。

Fig. 1-6 Structure of this thesis.

第一章：本章首先阐明了中文词法、句法、语义分析以及它们的联合模型的课题背景和意义，然后对中文的词法、句法以及语义分析的各自串行模型以及它们之间的联合模型的国内外研究现状进行了介绍，进一步，说明了联合模型所面临的前景和挑战，最后列出了本文的内容组织方式。

第二章：结合中文的特点，指出了中文的词具有内部结构，依托这样一个结构，我们可以将中文基于词的句法树转换成为基于字的句法树。如果我们使用一个模型直接对这样的字级别的句法树进行分析，便可以得到一个中文词法分析和句法分析的联合模型。我们将这个联合模型和相应的串行级联模型以及其它词法和句法的联合模型进行比较，发现我们的方法展现出更好的词法、句法分析性能，而且词内部结构也对中文词法句法的分析有着积极的帮助作用。

第三章：我们对一种关注度相对比较少的语义分析方法──语义依存分析──进行了说明和相关的比较分析，并且使用了另一种被广泛关注的浅层语义分析手段（语义角色标注）对语义依存分析进行了简单论述，以表明语义依存分析的有效性。紧接着，我们从理论以及实验分析两方面描述了句法依存特征对语义依存分析的帮助性，为语义依存分析和句法依存分析联合模型的提出做出了铺垫，最后我们提出了一种语义依存分析和句法依存分析的联合模型，实验结果表明了联合模型的方法能很好的提升语义依存分析的性能。

第四章：前人已经提出的词性标注和依存句法的联合模型可以分为三类，基于图的联合模型，基于转移的联合模型以及基于短语结构的联合模型。我们一方面通过将这三种不同类型的联合模型采用基于栈学习的方法进行融合，进一步提升了联合模型的性能；另一方面通过控制基于转移的联合模型中的柱的大小，结合大量未标注语料，采用过训练的方法提升了联合模型的速度而且保证了联合模型的高性能，最终我们得到了一个速度快而且性能高的词性标注和依存句法的联合模型。

第五章：目前最有效的提升统计模型在跨领域数据集上性能的最有效方法是标注少量目标领域的语料，然后和源领域语料结合在一起进行训练。但是基于同等的代价，如何标注这些目标领域的数据才能更有效呢? 这是一个非常值得研究的问题。针对该问题，我们提出将基于片段的词典标注和基于整体的句子标注相结合的方法来提升分词词性标注联合模型的领域自适应能力。我们的实验结果表明这种结合的方法比单纯的词典标注或者句子标注都要更有效。

# 第 2 章 基于字的词法句法联合模型

## 2.1 引言

汉语中的字在中文自然语言处理中占有非常重要的地位，它是最基本的输入单元。具有特定意义频繁出现的字序列往往会被认定为汉语中的词，目前大部分中文处理例如词性标注、句法分析以及语义分析都建立在词的基础上。但是实际上我们可以发现，大多数中文词是有内部句法结构的，例如图2-1中的“卧虎藏龙”，首先“卧”和“虎”，“藏”和“龙”分别作为一个偏正结构结合在一起，形成两个子词，然后这两个子词“卧虎”和“藏龙”进一步进行结合，构成一个并列结构的词。

分析中文词的内部结构，可以一定程度上减缓中文词在定义标准上的分歧。关于中文词的定义，存在着多种规范说明，单在BAKEOFF于2005年的评测中，便有四种不同的标注规范参与了中文分词评测[72]。如果词的内部句法结构也被分析出来，那么我们便可以利用一些规则来得到各种粒度不同的词，例如上面的成语“卧虎藏龙”中，“卧虎”和“藏龙”我们都可以看作一个词；再举一个例子，例如图2-3所示的“建筑业”，其内部结构中，首先“建”和“筑”构成一个并列结构子词，然后“建筑”和“业”再构成一个偏正结构的词，这样我们可以从它的内部结构中提取“建筑”和“业”两个词。

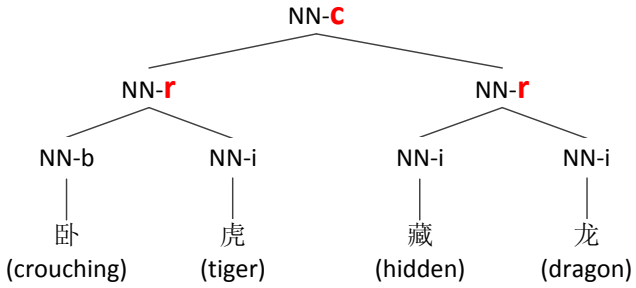
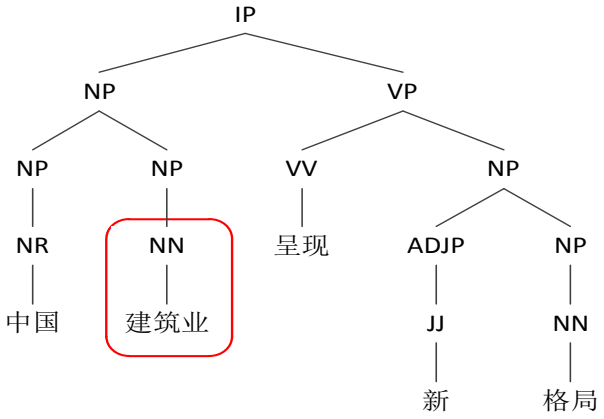



图 2-1 “卧虎藏龙”的内部句法结构。

Fig. 2-1 Character-level word structure of “卧虎藏龙(crouching tiger hidden dragon)”.

虽然词有结构，但是过去的研究工作往往忽略了这一点，图2-2显示了传统句法分析的一个例子，其中图2-2 a)是基于短语结构句法分析的，而图2-2 b)是基于依存结构句法分析的，这两种分析是在中文词的基础之上进行的结构分析，而且忽略了词的内部结构。

本章我们将研究词的内部结构对中文句子句法分析的影响，同时也希望中文词的内部句法结构能自动的被分析出来。为了达到上述两个目的，我们提出了基于字的句法分析，不仅能同时处理分词、词性标注以及句法分析等任务，而且能将中文词的内部句法结构分析出来。本章所涉及的句法分析包括两种主流的句法分析方法，即短语结构句法分析和依存结构句法分析。




a) 短语结构句法分析

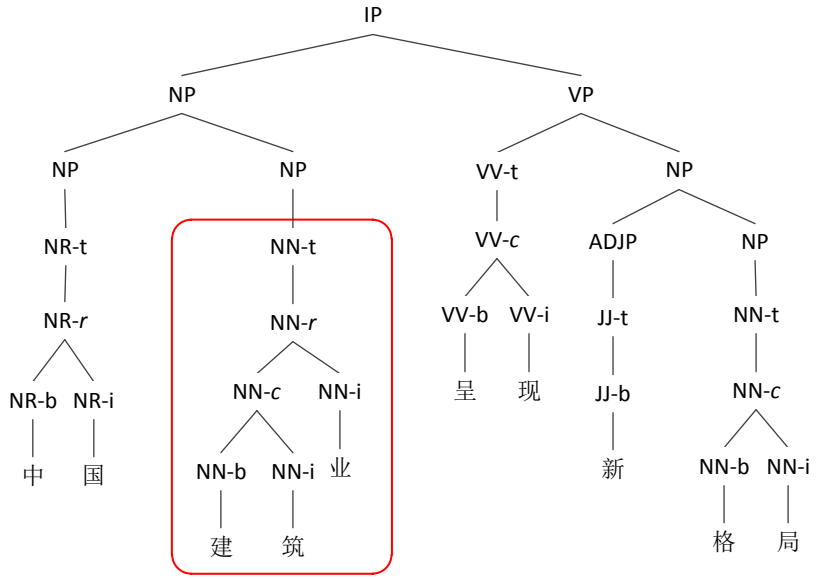

b) 依存结构句法分析

图 2-2 传统的基于词的中文句法分析

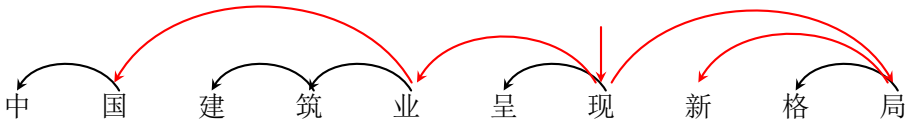
Fig. 2-2 Traditional word-based Chinese syntax parsing.

既然是基于字进行处理，那么一个高层的句子分析过程便不需要预先进行分词了，以及分词之后的词性标注也不需要了，也就是说我们可以用一个一体化的模型把传统中文自然语言处理的相关任务全部完成。在传统句法分析中，给定一个句子“中国建筑业呈现新格局”，其首先要进行分词，得到“中国 建筑业 呈现 新 格局”，然后在通过句法分析得到如图2-2所示的句法树。而如果我们基于字进行处理，便可以直接得到如图2-3所示的基于字的句法树，其中图2-3 a)为基于字的短语结构句法树，图2-3 b)为基于字的依存结构句法树。因此我们这种基于字的一体化的句法分析方法可以同时处理分词、词性标注以及句法分析任务。

使用一体化的方法来处理分词、词性标注以及句法分析，是近年来自然语言研究的一个热点。传统的方法是基于层次的，首先是进行分词，然后是词性




a) 短语结构句法分析




b) 依存结构句法分析

图 2-3 基于字的中文句法分析。

Fig. 2-3 Character-based Chinese syntax parsing.

标注，最后是句法分析。这种方法会引起错误传播的问题，比如分词的错误会导致词性标注的错误，词性标注的错误又会导致句法分析的错误，而一体化的分析方法会避免这个问题；同时一体化的分析方法还有一个优势，即上层所提供的信息能帮助下层的分析，例如句法分析的结果能帮助到词性标注，词性标注的结果也能帮助到分词。

## 2.2 相关工作

和本章最相关的研究工作是关于词内部结构分析的研究，最开始的工作是由赵海在2009年提出的基于字的依存结构句法分析[73]，他主要用这种基于字的分析方法来进行中文分词，同时词的内部结构也被自动地解析出来，后续的工作中他也将这种方法扩展到了句法分析，使得分词、词性标注以及依存结构句法分析一体化，由于模型选择和参数优化等等原因，他们并未取得较好的

性能。和他们的工作类似，本章中所标注的词内部结构也是基于字的，不同的是，我们所标注的词内部结构是基于短语句法结构的，并非基于依存结构，短语结构能比依存结构包含更多的信息，而且能转换成依存结构。我们将短语结构转换成依存结构后，用了一种更好的基于转移的柱搜索方式来进行解码，采用平均感知器与提前更新算法相结合来进行参数学习，从而取得了目前最好的性能，也验证了词内部结构的价值。

李中国等人也标注了中文词的内部结构[67, 70]，他们主要是进行词缀的标注工作，据统计，只有35%的中文词具有词缀结构，在除去词缀之后，他们的工作实际上也是基于词的，只不过词的粒度变的很细了；在这种细粒度词的基础之上，他们使用了一个联合模型将分词、词性标注和句法分析用一个统一的模型来解决，最初李中国基于短语结构句法分析来完成这一统一模型，但是性能并不理想。进一步，李中国和周国栋将该工作扩展到依存句法上面，最终的系统在句法分析上获得了比较大的提升，但是在分词和词性标注上性能有一定的损失。本章所提出的基于字的模型和他们的词缀结构是有很大区别的，而且我们提出的模型无论是在分词词性还是句法上都获得了一致的提升。

本文的方法主要是由基于转移的短语结构句法分析和基于转移的柱搜索依存结构句法分析扩展而得到的。在短语结构句法分析方面，王梦秋等人最先提出了基于转移的短语结构句法分析方法[24]，再进一步由张岳等人也将基于转移的短语结构句法分析和柱搜索相结合，从而得到了更好的句法分析性能[74]。在依存结构句法分析方面，基于转移的依存结构句法分析方法所采用的转移算法主要有两种，贪心弧转移算法和标准弧转移算法，在Nirve的文章中对这两种算法都有详细的介绍[35]，但是这篇文章中采用的是贪心搜索算法，而黄亮等人以及张岳等人分别将这两种转移算法与柱搜索相结合，采用全局打分方式，从而使得解码搜索空间变大[38, 39]，而且能够利用复杂的非局部特征。

联合模型的研究也与本章有着密切的关系，在基于短语句法的相关研究里，将分词、词性标注和短语结构句法分析进行融合的方法最早由罗小强提出[66]，他也采用了基于字的短语结构句法分析方法，不过其模型能融入的特征很有限，性能并不理想，而且从词到字，他采用的是一个伪造的内部词结构，没有实际的意义，只是为了完成一个联合分析而提出来的。Qian和Liu等人也提出了一个联合模型[68]，但是他们并不是使用一个统一的联合模型，而是先分别训练各自的概率模型，然后采用置信度信息进行联合解码。在依存句法方面，最为突出的工作是Hatori等人于2012年提出的联合模型[69]，他们将分词、词性标注和依存结构句法分析结合在一起，使用了标准弧转移算法，但是他们所提

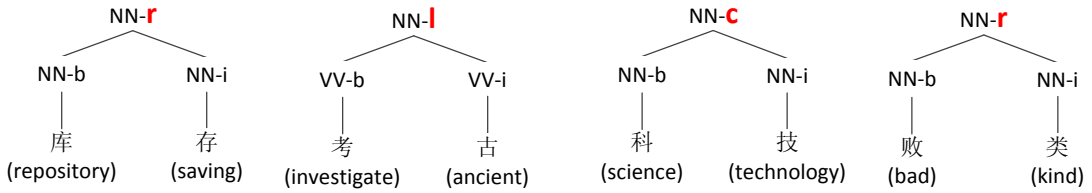
出的弧转移算法并不是非常完美。相比与前人提出的各种模型，我们取得了最好的性能，我们的模型扩展由于引入了词到字的结构，更直观更简洁。

## 2.3 词结构与基于字的中文句法树

在本小节中，我们介绍词结构的表示方法，主要和其它的词结构标注做一下区分，然后介绍词结构标注的规范以及数据规模，最后介绍如何扩展成为最终的基于字的句法树。

### 2.3.1 词结构表示

我们采用一棵完全二叉树来表示一个词，如图2-4所示为几个具体的例子，其中所举的这四个词，分别为主谓关系，动宾关系，并列关系以及修饰关系。在我们的标注中，我们标注了一个词是如何由更小的两个单位进行组合而形




a) 主谓关系　　b) 动宾关系　　c) 并列关系　　d) 修饰关系

图 2-4 “库存”，“考古”，“科技”和“败类”这四个词的内部结构。

Fig. 2-4 Inner word structures of “库存(repertory)”，“考古(archaeology)”，“科技(science and technology)”and “败类(degenerate)”.

成的。图2-5给了一个稍微复杂的例子，即一个包含了四个字的词。除了表明词的内部的合并组织方式，我们还指出了每一步合并时谁是核心部分，例如主谓关系中谓语为核心部分，动宾关系中动词为核心部分，并列关系中两部分同等重要，以及修饰关系中被修饰部分为核心部分，这个核心部分的指定分别用l（左，left），r（右，right）以及c（并列，coordinate）来表明。最下面一层的非叶子节点，表明了这个字处于其所在词中的位置，简单区分为两种，一个是开始位置(begin)，用b表示，另一个是非开始位置，用i表示。

李中国等人也标注了一个基于词缀的词结构，我们的词结构标注和他们的标注有着很大的区别。他们仅仅标注了词缀，根据他们的统计，含有词缀的数目占词总数的35%左右，而我们对所有词，所有部分结构都做了标注，如图2-6显示了几个例子。

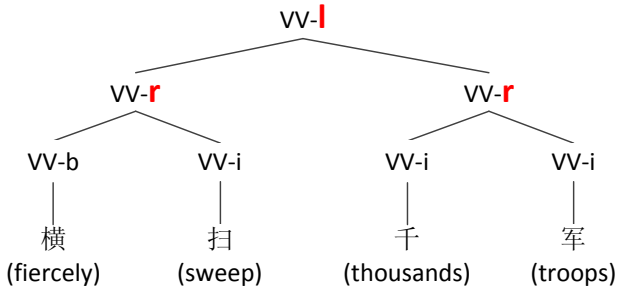



图 2-5 “横扫千军”的内部句法结构。

Fig. 2-5 Character-level word structure of “横扫千军(make a clean sweep of millions troops)”.

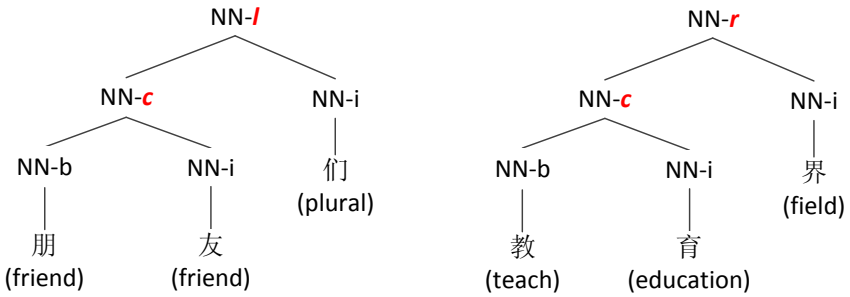
### 2.3.2 词结构标注

我们的词结构是由三个人手工标注的，在大部分情况下，词语内部组成方式还是比较容易标注的，最大的难点在于父亲节点的指定，我们综合借鉴了常用的依存句法标注规范，总结出了以下若干规则：

1. 动词+名词，表示动宾含义时，核心在动词上面，如果表示修饰关系，则核心在名词上面；
2. 动词+动词，两者都是普通动词时，并列关系，如果后面的动词是形容词类动词，比如可以用”很”之类的词修饰，则核心在左边动词上面，如果后面动词为“有”或者“是”等词时，核心在后面词上面；
3. 形容词+名词，基本上都表示修饰关系，重核心在名词上面；
4. 名词+名词，如果表示重复表示同一物体或者表示两个相关联的物体，标为并列结构；如果表示同一物体但是前者后者意义有差别，一般前者是修饰后者时，核心在后面词上面；
5. 介词+名词，或者介词+动词，核心在介词上面；
6. 动词/形容词+ 副词，核心在前面词上面；
7. 名词+动词，表示主谓关系时，核心在动词上面；
8. 主谓宾结合的情况，主谓先结合，然后再和宾语结合；
9. 子词+前/后/左/右/上/下/中，表示方位时间时，核心在后面；
10. 数词+数词，英文字母也类似处理，统一对待，全部表并列，内部结构不区分；
11. 名字，名和姓分开。
12. 对于上面规则无法覆盖的部分，使用并列表示其内部结构。

最终我们分两阶段标注完了中文宾州句法树库(Chinese Penn Treebank,


a) 基于词缀的标注




b) 基于字的标注

图 2-6 我们的标注方法和基于词缀的标注方法的一个对比。

Fig. 2-6 Comparison between character-level and morphological-level word structures.

CTB)7.0版本中的所有词，第一阶段标注完四万多词，第二阶段又标注了两万词左右，最终总计标注了67,197个词。

### 2.3.3 基于字的短语结构句法树

前面图2-2 a)显示了传统的基于词的短语结构句法树的一个例子，而在图2-4以及在图2-5中我们给出了词内部结构标注的例子，从里面能看出，基于词的短语结构句法树要转换成基于字的短语结构句法结构（如图2-3 a)所示），是非常容易的，只需要把词的内部结构扩展插入到基于词的短语结构句法树中。为了分开词以下的子词结构节点和词以上的短语结构节点，中间我们插入了一层词结构节点，以t来标记这个特殊的节点。由于我们对CTB短语树库中所有的词的结构都进行了标注，所以我们可以非常方便的将CTB中所有的基于词的短语结构转换成基于字的短语结构。

### 2.3.4 基于字的依存结构句法树

图2-2 b)显示了传统的基于词的依存结构句法分析的一个例子。由于我们所标注的词内部结构是基于短语结构的，而如果想把基于词的依存结构句法树

转换成基于字的依存结构树，我们需要将词内部结构转化为采用依存结构表示。实际上，我们在标注词内部结构时已经考虑到了这一点，因为任何词内部节点都包括两个子树，而且其中一棵子树是中心（对于并列结构，我们自动使用右边的子树为中心），因此我们标注的所有词内部结构，都可以非常容易的转换成为字依存结构，进而把这个依存结构附加在基于词的依存句法树上便形成了基于字的依存结构句法树，如图2-3 b)所示。

## 2.4 基于字的短语结构句法分析模型

我们采用基于转移的方法，通过扩展张岳和Clark在2009年提出的基于转移的短语结构句法分析[25, 74]，来实现基于字的短语结构句法分析模型。在基于转移的系统中，状态和转移动作定义是最核心的内容。在基于字的短语结构句法分析模型中，其状态定义与基于词的短语结构句法分析类似，如图2-7所示，而其转移操作一共有七类，这七类动作能生成任意的度小于等于2的句法树。实际的短语句法树中，可能会有很多度大于2的节点，我们通过一定的父亲节点发现规则自动将多元句法树转换成二元句法树，在转换过程中我们引入了一些特殊标记，使得转换后的二元句法树能恢复到原来的句法树。具体这七类转移动作的定义分别如下所示。

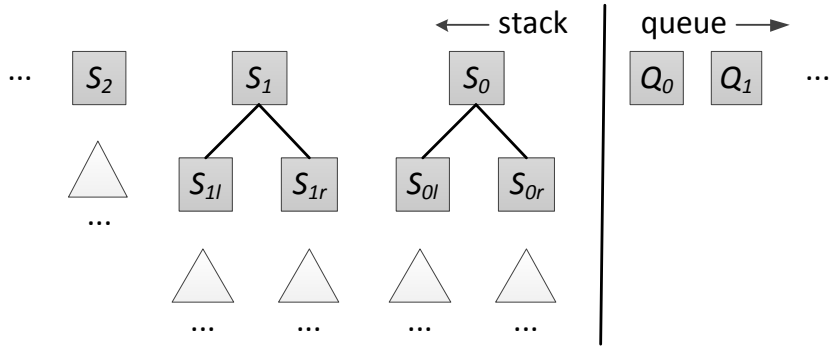



图 2-7 基于字的短语结构句法分析模型中状态的定义。

Fig. 2-7 State definition of the character-based Chinese constituent parsing.

1. SHIFT-SEPARATE($t$)：将队列$Q$中的第一个字$c_j$移出，形成一个子词(subword)结构节点$\frac{S'}{c_j}$[1]并移入栈$S$中，同时为该字赋予词性$S'.t = t$，需要注意的是此时$S_0$必须为一个完整的词或者短语结构节点，不能为子词结构节点。

[1]我们借此简要表示一个二叉结构树节点，其中分母代表孩子节点，分子代表父亲节点

2. SHIFT-APPEND：将队列$Q$中的第一个字$c_j$移出，形成一个子词(subword)结构节点$\frac{S'}{c_j}$并移入栈$S$中。$c_j$最终将和栈$S$中最顶端的几个子词结构节点结合，形成一个完整的词，因此这个SHIFT操作没有额外的参数，词性早已由该词的第一个字在被移入时指定。
3. REDUCE-SUBWORD($d$)：将栈顶的两个子词结构节点$S_0$和$S_1$合并，形成一个更大的子词结构节点$\frac{S'}{S_1\ S_0}$并置于栈顶，其中参数$d$表示这个合并节点$S'$的核心方向，其值可以是“左”，“右”或者“并列”。
4. REDUCE-WORD：将栈顶的子词结构节点转换成一个完整词结构节点$\frac{S'}{S_0}$。
5. REDUCE-BINARY($d,l$)：将栈顶的两个节点$S_0$ and $S_1$合并并形成一个新的短语结构节点$\frac{S'}{S_1\ S_0}$置于栈顶，其中参数$l$表示$S'$的成分标签，参数$d$指明两个子树节点谁是核心，其值可以是“左”或者“右”。$S_0$和$S_1$必须都是完整词结构节点或者短语结构节点，不能是子词结构节点。
6. REDUCE-UNARY($l$)：将栈顶的完整词结构节点或者短语结构节点转换成一个一元的短语结构子树节点$\frac{S'}{S_0}$并置于栈顶，其中参数$l$表示$S'$的标签。
7. TERMINATE：如果栈中只剩一棵子树，队列中没有任何未处理的字，则句子短语结构句法树已经分析完毕，我们用这一操作标记分析结束。

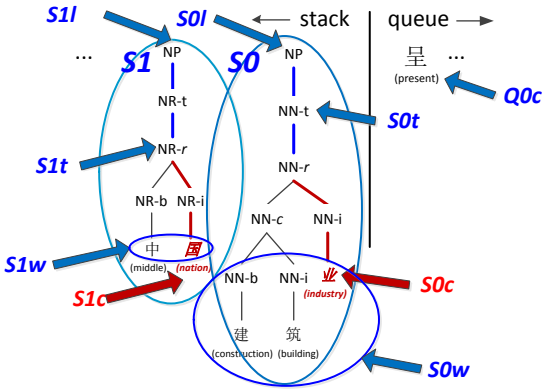



图 2-8 基于字的短语结构分析模型所用到的特征元素。

Fig. 2-8 Atomic features of the character-based constituent parsing.

在这个基于字的句法分析系统中，所使用的特征如表2-1所示，其中各个符号的含义如图2-8所示，函数start(·), end(·) 以及len(·) 分别代表一个词的第一个字，最后一个字以及包含字的总个数。表中提到的特征包括传统的分词词性联合模型的特征[3]，如其中的线性特征所示，也包括基于词的短语结构句法分析的特征（结构特征）[74]，其中我们还增加了字结构信息带来的字特征，如表格中的粗体部分所示。

表 2-1 基于字的句法分析模型中所使用的特征。

Table 2-1 Features templates of the character-based constituent parsing model.

| 类别 | 特征模板 |
|---|---|
| 结构特征 | $S_0ntl$, $S_0nwl$, $S_1ntl$, $S_1nwl$, $S_2ntl$, $S_2nwl$, $S_3ntl$, $S_3nwl$ |
| | $Q_0c$, $Q_1c$, $Q_2c$, $Q_3c$ $Q_0c \circ Q_1c$, $Q_1c \circ Q_2c$, $Q_2c \circ Q_3c$ |
| | $S_{0l}twl$, $S_{0r}twl$, $S_{0u}twl$, $S_{1l}twl$, $S_{1r}twl$, $S_{1u}twl$ |
| | $S_0nw \circ S_1nw$, $S_0nw \circ S_1nl$, $S_0nl \circ S_1nw$, $S_0nl \circ S_1nl$ |
| | $S_0nw \circ Q_0c$, $S_0nl \circ Q_0c$, $S_1nw \circ Q_0c$, $S_1nlQ_0c$ |
| | $S_0nl \circ S_1nl \circ S_2nl$, $S_0nw \circ S_1nl \circ S_2nl$ |
| | $S_0nl \circ S_1nw \circ S_2nl$, $S_0nl \circ S_1nl \circ S_2nw$ |
| | $S_0nw \circ S_1nl \circ Q_0c$, $S_0nl \circ S_1nw \circ Q_0c$, $S_0nl \circ S_1nl \circ Q_0c$ |
| | **$S_0ncl$, $S_0nct$, $S_0nctl$, $S_1ncl$, $S_1nct$, $S_1nctl$** |
| | **$S_2ncl$, $S_2nct$, $S_2nctl$, $S_3ncl$, $S_3nct$, $S_3nctl$** |
| | **$S_0nc \circ S_1nc$, $S_0ncl \circ S_1nl$, $S_0nl \circ S_1ncl$, $S_0ncl \circ S_1ncl$** |
| | **$S_0nc \circ Q_0c$, $S_0nl \circ Q_0c$, $S_1nc \circ Q_0c$, $S_1nl \circ Q_0c$** |
| | **$S_0nc \circ S_1nc \circ Q_0c$, $S_0nc \circ S_1nc \circ Q_0c \circ Q_1c$** |
| | start($S_0w$) $\circ$ start($S_1w$), start($S_0w$) $\circ$ end($S_1w$) |
| | indict($S_1wS_0w$) $\circ$ len($S_1wS_0w$) indict($S_1wS_0w, S_0t$) $\circ$ len($S_1wS_0w$) |
| 线性特征 | $t_{-1}t_0$, $t_{-2}t_{-1}t_0$, $w_{-1}t_0$, $c_0t_0$ start($w_{-1}$)$t_0$, $c_{-1}c_0t_{-1}t_0$ |
| | $w_{-1}$, $w_{-2}w_{-1}$, $w_{-1}$, where len($w_{-1}$) = 1, end($w_{-1}$)$c_0$ |
| | start($w_{-1}$) $\circ$ len($w_{-1}$), end($w_{-1}$) $\circ$ len($w_{-1}$), start($w_{-1}$) $\circ$ end($w_{-1}$) |
| | $w_{-1}c_0$, end($w_{-2}$) $\circ$ $w_{-1}$, start($w_{-1}$)$c_0$, end($w_{-2}$) $\circ$ end($w_{-1}$) |
| | $w_{-1} \circ$ len($w_{-2}$), $w_{-2} \circ$ len($w_{-1}$), $w_{-1}t_{-1}$, $w_{-1}t_{-2}$, $w_{-1}t_{-1}c_0$ |
| | $w_{-1}t_{-1} \circ$ end($w_{-2}$), $c_{-2}c_{-1}c_0t_{-1}$, where len($w_{-1}$) = 1, end($w_{-1}$)$t_{-1}$ |
| | $ct_{-1} \circ$ end($w_{-1}$), where $c \in w_{-1}$ *and* $c \neq$ end($w_{-1}$) |
| | $c_0t_{-1}$, $c_{-1}c_0$, start($w_{-1}$)$c_0t_{-1}$, $c_{-1}c_0t_{-1}$ |

## 2.5 基于字的依存结构句法分析模型

和基于字的短语结构句法分析类似，我们也使用基于转移的解码算法结合柱搜索来实现基于字的依存结构句法分析，我们通过扩展传统词级别的基于转移的依存结构句法分析来实现字级别的依存结构句法分析。基于词的依存结构句法分析方法有两种典型的转移算法，一种是标准弧转移算法，另一种是贪心弧转移算法，这两种算法都在Nirve等人的论文中有详细的介绍[35]。在本节中我们将分别对这两种算法进行扩展，得到基于字的标准弧转移算法和贪心弧转移算法。

实际上我们也可以完全使用原来的基于词的依存结构句法分析方法来处理字依存句法树，即把每个字都当作一个独立的词一样对待，但是这样做我们便无法使用前人工作中已经提出的基于词的一些特征，过去的不少研究工作都已经表明这些特征的有效性。因此，对基于词的依存结构句法分析进行一定的改进是非常必要的，使得既能使用基于词的特征，又能使用一些字相关的特征以及根据词内部结构所提出来的特征，下面我们将分别介绍这两种扩展。

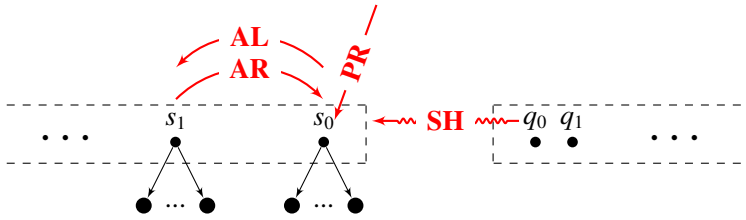
### 2.5.1 标准弧转移算法

首先我们介绍基于词的标准弧转移算法，然后我们再介绍如何将该算法扩展为基于字的依存分析算法，对于标准弧转移算法，每个状态由一个栈和一个队列构成，栈中存储的是部分解码的依存句法树序列，而队列中存储的是尚未处理的词序列，如图2-9 a)所示，其中$s_0$和$s_1$表示栈顶的第一棵和第二棵部分解码依存树，$q_0$和$q_1$表示队列中的第一个和第二个词。在该算法中，我们定义了四种不同的状态转移操作，分别是移进（SHIFT，SH），左弧（ARC-LEFT，AL），右弧（ARC-RIGHT，AR）和移除根节点（POP-ROOT，PR），如图2-9 a)所示，进一步我们对该四种操作进行详细解释：

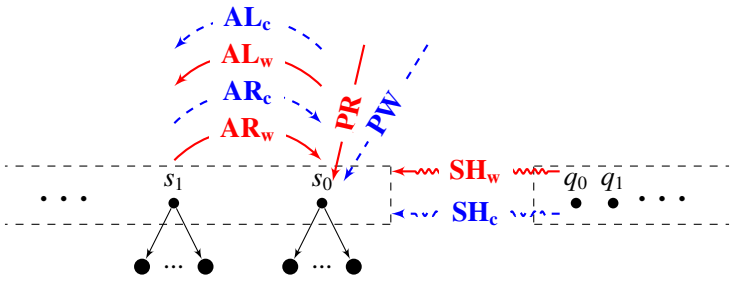
- SHIFT：将队列中的第一个元素移入栈中，形成一个仅包含一个节点的部分解码依存树；
- ARC-LEFT：将栈顶的两棵部分解码依存树进行合并，形成一个左弧，使得栈顶的第一棵依存树的根节点为合并后的新的部分解码依存树的根节点；
- ARC-RIGHT：将栈顶的两棵部分解码依存树进行合并，形成一个右弧，使得栈顶的第二棵依存树的根节点为合并后的新的部分解码依存树的根

节点；

- POP-ROOT：如果队列中的元素为空，而且栈中只包含一棵部分解码依存树时，将这棵依存树移除，并令这棵依存树的根节点为全句分析的根节点。




a) 基于词的依存分析




b) 基于字的依存分析

图 2-9 使用标准弧转移算法时，基于词的和基于字的依存结构句法分析模型对比。

Fig. 2-9 A comparison between word-based and character-level arc-standard algorithms.

将上述基于词的依存分析算法扩展到基于字的依存分析时，我们对原有的状态定义不加改变（栈中存储的是字依存树序列，而队列中存储的是尚未处理的字序列），而是简单的扩充转移操作，将转移操作扩充为两类，一类是面向词的，也就是和原有的基于词的分析算法所起的功能一样，另一类是面向字的，也就是在成词之前的状态转移用这些操作。这两类操作完全类似，分别为词移进（$\text{SHIFT}_w$，$\text{SH}_w$），词左弧（$\text{ARC-LEFT}_w$，AL），词右弧（$\text{ARC-RIGHT}_w$，$\text{AR}_w$），移除根节点（POP-ROOT，PR）以及字移进（$\text{SHIFT}_c$，$\text{SH}_c$），字左弧（$\text{ARC-LEFT}_c$，$\text{AL}_c$），字右弧（$\text{ARC-RIGHT}_c$，$\text{AR}_c$），词形成（POP-WORD，PW），下面我们分别对这八种操作加以解释。

- $\text{SHIFT}_w$：将队列中的第一个元素移入栈中，形成一个仅包含一个节点的部分解码字依存树，该字将会成为一个中文词的第一个字，该操作有一个参数，表示一个词的词性；
- $\text{ARC-LEFT}_w$：将栈顶的两个部分解码字依存树进行合并，形成一个左

弧，使得栈顶的第一棵依存树的根节点为合并后的新的部分解码依存树的根节点，参与合并的两个部分解码字依存树中，所有词已经确定；

- ARC-RIGHT$_w$：将栈顶的两棵部分解码依存树进行合并，形成一个右弧，使得栈顶的第二棵依存树的根节点为合并后的新的部分解码依存树的根节点，参与合并的两个部分解码字依存树中，所有词已经确定；
- POP-ROOT：如果队列中的元素为空，而且栈中只包含一棵部分解码依存树时，将这棵依存树移除，并令这棵依存树的根节点为全句分析的根节点；
- SHIFT$_c$：将队列中的第一个元素移入栈中，形成一个仅包含一个节点的部分解码字依存树；
- ARC-LEFT$_c$：将栈顶的两个部分解码字依存树进行合并，形成一个左弧，使得栈顶的第一棵依存树的根节点为合并后的新的部分解码依存树的根节点，参与合并的两个部分解码字依存树中，词都没有形成；
- ARC-RIGHT$_c$：将栈顶的两棵部分解码依存树进行合并，形成一个右弧，使得栈顶的第二棵依存树的根节点为合并后的新的部分解码依存树的根节点，参与合并的两个部分解码字依存树中，词都没有形成；
- POP-WORD：标记栈中最顶元素的字依存树为一个词，或者说最顶元素的字依存树构成一个词。

进一步，我们介绍这个基于字的依存分析系统中所使用的特征，一共分为三部分，其中两部分是前人提出的分词词性标注联合模型中所使用的特征和传统的基于词的依存句法分析特征，如表2-2所示；第三部分为新提出的融入了词内部结构的特征，如表2-3所示。其中$c$，$w$和$t$分别代表字，词和词性；$S$和$Q$代表栈和队列；$L$和$R$代表建立弧的两棵部分解码依存树所在的相对位置；下标中的数字表示距离此处分析所在位置的距离；$s(w)$，$e(w)$以及$l(w)$ 表示某个词的第一个字，最后一个字和该词的长度；$lc1$，$lc2$，$rc1$和$rc2$分别表示最左边的第一个子孩子，最左边的第二个孩子，最右边的第一个子孩子和最右边的第二个孩子；$lval$和$rval$表示一棵依存树的左边孩子数目和右边孩子数目；$\underline{lsw}$ and $\underline{rsw}$ 表示一个词的最小左子词和最小右子词，具体的例子如图2-10所示。

### 2.5.2 贪心弧转移算法

和上节组织方式类似，首先我们介绍基于词的贪心弧转移算法，然后我们再介绍如何将该算法扩展为基于字的依存分析算法，对于贪心弧转移算

表 2-2 前人提出的特征。

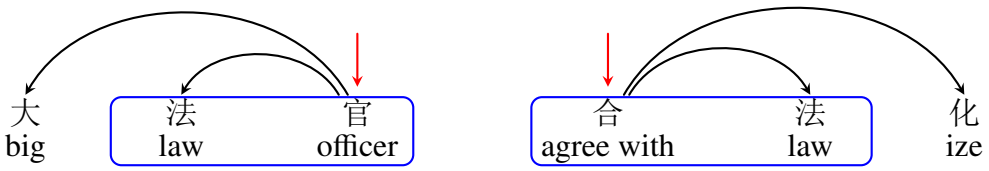
Table 2-2 The feature templates proposed by previous work.

| 使用特征的动作 | 特征模板 |
|---|---|
| 分词词性标注 | |
| 移入词的开始字 | $t_{-1}t_0$, $t_{-2}t_{-1}t_0$, $w_{-1}t_0$, $c_0t_0$, $s(w_{-1})\circ t_0$, $c_{-1}c_0t_{-1}t_0$, |
| 词形成 | $w_{-2}w_{-1}$, $e(w_{-1})\circ t_{-1}$, $e(w_{-1})\circ c_0$, $w_{-1}c_0$, $w_{-1}t_{-1}\circ e(w_{-2})$, $w_{-1}$, $s(w_{-1})\circ l(w_{-1})$, $e(w_{-1})\circ l(w_{-1})$, $s(w_{-1})\circ e(w_{-1})$, $e(w_{-2})\circ e(w_{-1})$, $e(w_{-2})\circ w_{-1}$, $s(w_{-1})\circ c_0$, $w_{-1}\circ l(w_{-2})$, $w_{-2}\circ l(w_{-1})$, $w_{-1}t_{-2}$, $w_{-1}t_{-1}$, $w_{-1}t_{-1}c_0$, $w_{-1}$, where $l(w_{-1})=1$, $c_{-2}c_{-1}c_0t_{-1}$, where $l(w_{-1})=1$, $ct_{-1}\circ e(w_{-1})$, where $c\in w_{-1}$ , |
| 移入词的非开始字 | $c_0t_{-1}$, $c_{-1}c_0$, $c_{-1}c_0t_{-1}$, $s(w_{-1})\circ c_0t_{-1}$ |
| 依存结构句法分析 | |
| 句法相关的动作 | $S_0w$, $S_0t$, $S_0wt$, $S_1w$, $S_1t$, $S_1wt$, $Q_0w$, $S_0w\circ S_1w$, $S_0t\circ S_1t$ $S_0w\circ S_1t$, $S_0t\circ S_1w$, $S_0wt\circ S_1t$, $S_0t\circ S_1wt$ $S_0wt\circ S_1w$, $S_0w\circ S_1wt$, $S_0wt\circ S_1wt$ $S_0t\circ S_1t\circ S_{1lc}t$, $S_0t\circ S_1t\circ S_{1rc}t$, $S_0w\circ S_1t\circ S_{1rc}t$ $S_0w\circ S_1t\circ S_{1rc}t$, $S_0t\circ S_1t\circ S_{0lc}t$, $S_0t\circ S_1t\circ S_{0rc}t$ $S_0w\circ \mathrm{dist}$, $S_0t\circ \mathrm{dist}$, $S_1w\circ \mathrm{dist}$, $S_1t\circ \mathrm{dist}$ $S_0w\circ \mathrm{lval}(S0)$, $S_0t\circ \mathrm{lval}(S0)$ $S_1w\circ \mathrm{lval}(S1)$, $S_1t\circ \mathrm{lval}(S1)$, $S_1w\circ \mathrm{rval}(S1)$, $S_1t\circ \mathrm{rval}(S1)$ $S_{0lc}w$, $S_{0lc}t$, $S_{1lc}w$, $S_{1lc}t$, $S_{1rc}w$, $S_{1rc}t$ $S_{0lc2}w$, $S_{0lc2}t$, $S_{1lc}w$, $S_{1lc2}t$, $S_{1rc2}w$, $S_{1rc2}t$ $S_0t\circ S_{0lc}t\circ S_{0lc2}t$, $S_1t\circ S_{1lc}t\circ S_{1lc2}t$, $S_1t\circ S_{1rc}t\circ S_{1rc2}t$ |

表 2-3 新提出的融入了词内部结构的特征。

Table 2-3 The new feature templates encoding inner-word dependencies.

| 特征模板 |
|---|
| $L\underline{c}$, $L\underline{c}\underline{t}$, $R\underline{c}$, $R\underline{c}\underline{t}$, $L_{lc1}\underline{c}$, $L_{rc1}\underline{c}$, $R_{lc1}\underline{c}$, $L\underline{c}\circ R\underline{c}$, $L_{lc1}\underline{c}\underline{t}$, $L_{rc1}\underline{c}\underline{t}$, $R_{lc1}\underline{c}\underline{t}$, $L\underline{c}\circ R\underline{w}$, $L\underline{w}\circ R\underline{c}$, $L\underline{c}\underline{t}\circ R\underline{w}$, $L\underline{w}\underline{t}\circ R\underline{c}$, $L\underline{w}\circ R\underline{c}\underline{t}$, $L\underline{c}\circ R\underline{w}\underline{t}$, $L\underline{c}\circ R\underline{c}\circ L_{lc1}\underline{c}$, $L\underline{c}\circ R\underline{c}\circ L_{rc1}\underline{c}$, $L\underline{c}\circ R\underline{c}\circ L_{lc2}\underline{c}$, $L\underline{c}\circ R\underline{c}\circ L_{rc2}\underline{c}$, $L\underline{c}\circ R\underline{c}\circ R_{lc1}\underline{c}$, $L\underline{c}\circ R\underline{c}\circ R_{lc2}\underline{c}$, $Ll\underline{sw}$, $Lr\underline{sw}$, $Rl\underline{sw}$, $Rr\underline{sw}$, $Ll\underline{swt}$, $Lr\underline{swt}$, $Rl\underline{swt}$, $Rr\underline{swt}$, $Ll\underline{sw}\circ R\underline{w}$, $Lr\underline{sw}\circ R\underline{w}$, $L\underline{w}\circ Rl\underline{sw}$, $L\underline{w}\circ Rr\underline{sw}$ |




a) 最小左子词　　b) 最小右子词

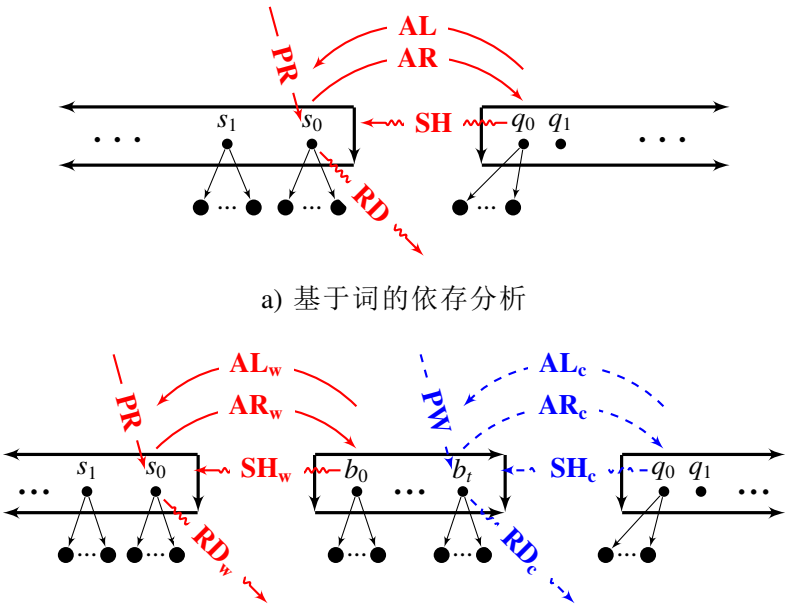
图 2-10 最小左子词和最小右子词的一个例子。

Fig. 2-10 An example to illustrate the innermost left/right subwords.

法，每个状态也是由一个栈和一个队列构成，栈中存储的是部分解析的依存句法树序列，而队列中存储的是尚未处理的词序列，除了第一个字的部分右子树已经找到，如图2-11 a)所示，其中$s_0$和$s_1$表示栈顶的第一个和第二棵部分解码依存树，$q_0$和$q_1$表示队列中的第一个元素和第二个元素，$q_0$也可能是一个部分解码树。在该算法中，我们定义了五种不同的状态转移操作，分别是移进（SHIFT，SH），左弧（ARC-LEFT，AL），右弧（ARC-RIGHT，AR）归约（REDUCE，RD）和移除根节点（POP-ROOT，PR），如图2-11 a)所示，进一步我们对该五种操作进行详细解释：

- SHIFT：将队列中的第一个元素移入栈中；
- ARC-LEFT：将栈顶的部分解码依存树和队列中的第一个元素进行合并，形成一个左弧，使得队列中的第一个元素的根节点为合并后的新的部分解码依存树的根节点，同时将栈顶的部分解码依存树移出栈；
- ARC-RIGHT：将栈顶的部分解码依存树和队列中的第一个元素进行合并，形成一个右弧，使得栈顶的部分解码依存树的根节点为合并后的新的部分解码依存树的根节点，同时将队列中的第一个元素移入栈中；
- REDUCE：将栈顶的部分解码依存树移出栈；
- POP-ROOT：如果队列中的元素为空，而且栈中只包含一棵部分解码依存树时，将这棵树移除，并令这棵依存树的根节点为全句的根节点。

将上述基于词的贪心弧转移依存分析算法扩展到基于字的依存分析时，我们不仅要对原有的操作进行扩充，而且状态的定义也需要改变。对于标准弧转移算法，队列中的元素移入栈只发生在SHIFT操作中，而对于贪心弧转移算法，情况有一些变化，ARC-RIGHT也能将队列中的元素移入栈中，这样带来了词性赋予的不方便性，不过我们对状态的定义稍加扩充，便可以解决这一问题。我们定义了一个缓冲区，其具体形式如图2-11 b)所示缓冲区中存储着若干字依存




a) 基于词的依存分析

b) 基于字的依存分析

图 2-11 使用贪心弧转移算法时，基于词的和基于字的依存结构句法分析模型对比。

Fig. 2-11 A comparison between word-based and character-level arc-eager algorithms.

树，除了缓冲区靠近队列那一端的几个连续元素词未形成之外，其他的元素都已经被标记成词。在实际进行状态转移变换时，词基础上的转移操作都在栈和缓冲区之间进行，而词内部的转移操作都在缓冲区和队列中运行。在进行词基础上的操作时，所有参与词的词性已经指定；在进行词内部的操作时，每个词相当于建立一棵依存树，只需要首字移入时赋予词性，其他的字不需要赋予词性，也就是说每建立一棵依存树，只需要固定的一次赋予词性操作，因此避免在何时进行词性赋值这个问题。

同样，我们也对转移操作也加以扩充，将其扩充为两类，一类是面向词的，也就是和原有的基于词的分析算法所起的功能一样，另一类是面向字的，也就是在成词之前的状态转移用这些操作。这两类操作完全类似，分别为词移进（$\text{SHIFT}_w$，$\text{SH}_w$），词左弧（$\text{ARC-LEFT}_w$，AL），词右弧（$\text{ARC-RIGHT}_w$，$\text{AR}_w$），词归约（$\text{REDUCE}_w$，$\text{RD}_w$），移除根节点（POP-ROOT，PR）以及字移进（$\text{SHIFT}_c$，$\text{SH}_c$），字左弧（$\text{ARC-LEFT}_c$，$\text{AL}_c$），字右弧（$\text{ARC-RIGHT}_c$，$\text{AR}_c$），字归约（$\text{REDUCE}_c$，$\text{RD}_c$），词形成（POP-WORD，PW），下面我们分别对这八种操作加以解释：

- $\text{SHIFT}_w$：将缓冲区中的第一个元素移入栈中；

- ARC-LEFT$_w$：将栈顶的部分解码依存树和缓冲区中的第一个元素进行合并，形成一个左弧，使得缓冲区中的第一个元素的根节点为合并后的新的部分解码依存树的根节点，同时将栈顶的部分解码依存树移出栈；
- ARC-RIGHT$_w$：将栈顶的部分解码依存树和缓冲区中的第一个元素进行合并，形成一个右弧，使得栈顶的部分解码依存树的根节点为合并后的新的部分解码依存树的根节点，同时将缓冲区中的第一个元素移入栈中；
- REDUCE$_w$：将栈顶的部分解码依存树移出栈；
- POP-ROOT：如果队列中的元素以及缓冲区中的元素都为空，而且栈中只包含一棵部分解码依存树时，将这棵依存树移除，并令这棵依存树的根节点为全句分析的根节点。
- SHIFT$_c$：将队列中的第一个元素移入缓冲区中；
- ARC-LEFT$_c$：将缓冲区靠近队列一侧的部分解码依存树和队列中的第一个元素进行合并，形成一个左弧，使得队列中的第一个元素的根节点为合并后的新的部分解码依存树的根节点，同时将栈顶的部分解码依存树移出栈；
- ARC-RIGHT$_c$：将缓冲区靠近队列一侧的部分解码依存树和队列中的第一个元素进行合并，形成一个右弧，使得缓冲区靠近队列一侧的部分解码依存树的根节点为合并后的新的部分解码依存树的根节点，同时将队列中的第一个元素移入缓冲区中；
- REDUCE$_c$：将缓冲区靠近队列一侧的部分解码依存树移出栈；
- POP-WORD：标记缓冲区靠近队列一侧的部分解码字依存树为一个词。

前面介绍了算法的状态以及状态转移操作，其具体解码方式以及训练方式和标准弧转移算法类似，不再重复介绍。其所用到的特征也分为三类，第一类是传统的分词和词性标注联合模型中所用到的特征，第二类是传统的基于词的贪心弧转移依存算法中所用到的特征，第三类是新提出的融入了内部词结构信息的特征。其中第一类和第三类和标准弧转移算法一模一样，第二类特征和标准弧转移算法也基本一样，但是需要将原来的$Q_0$、$S_0$和$S_1$分别变换为$Q_1$、$Q_0$和$S_0$。

## 2.6 实验结果与分析

### 2.6.1 实验设置

为了测试和比较这种基于字的句法分析性能，我们在CTB 5.0的数据集上进行实验，表2-4给出了实验室数据的统计信息。我们对CTB中的中文句法结构做了类似于Mary Harper和Zhongqiang Huang 在2011 年的预处理[75]，这个预处理主要是将多层的一元句法规则转换成单层的一元句法规则。为了将CTB中的所有句法树转换成度小于2的树，我们使用了张岳和Clark在2010年提出的父亲节点发现规则[38]。

表 2-4 语料划分以及相关信息.

Table 2-4 Corpus statistics.

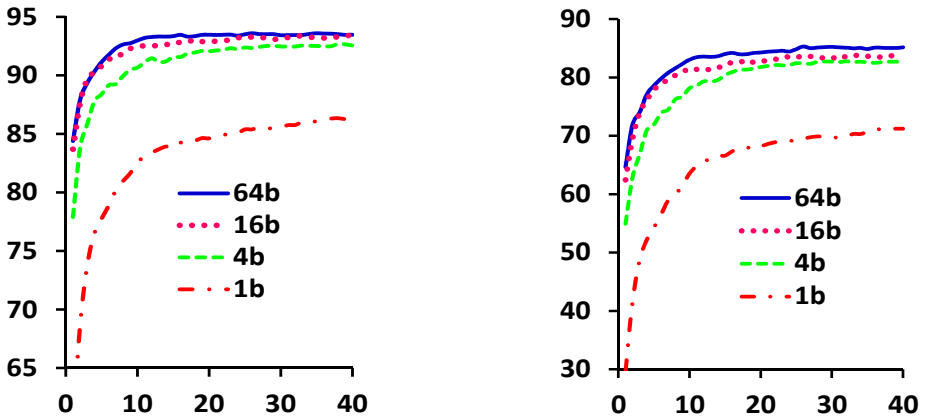
| | 划分 | 句子数目 | 词数目 |
|---|---|---|---|
| 训练集 | 001–270;<br>400–1151; | 18,089 | 493,939 |
| 开发集 | 301–325; | 350 | 6,821 |
| 测试集 | 271–300; | 348 | 8,008 |

对于基于字的短语结构句法分析，由于我们的方法能同时分析出分词、词性标注和短语结构句法分析的结果，所以我们至少有三种评价方法。首先分词的评价，我们采用分词的准确率(P)，即分词结果中正确的词数占自动分析结果中的总词数的比例、分词的召回率(R)，即分词结果中正确的词数占正确结果中总词数的比例、以及分词的F值，F值的计算公式为$\frac{2PR}{P+R}$；其次词性标注的评价，我们采用基于词的评价方式，包括词性标注准确率(P)，即词性标注结果中词性标注正确的词数占自动分析结果中的总词数的比例、词性标注召回率(R)，即词性标注结果中词性标注正确的词数占正确结果中总词数的比例、以及词性标注的F值；然后句法的评价，我们的评价方式和前人的评价方式例如Xian Qian和Yang Liu 在2012 年提出的方法一致[68]，识别句法树中短语结构节点的准确率(P)、召回率(R)和F值，一个短语被正确识别的条件是其开始边界、结束边界以及短语结构标签都识别正确，这个边界根据其叶子层节点最两端的字在句子中出现的位置计算。除了这三个评价指标之外，我们还能提供词结构分析的性能，类似于分词和词性标注，我们也采用了基于词的评价方法，如果一个词的内部结构树全部被正确识别，则表明该词的词结构分析正确（不考虑词性）。

对于基于字的依存结构句法分析，我们也能分析出分词，词性标注，同时还能得到依存结构句法分析的结果，因此我们也分别评价了分词、词性标注和依存结构句法分析的性能。只有当一条依存弧两边的词在分词正确而且这条弧的指向也正确的情况下，我们才认为这条依存弧正确。类似，我们也使用了词内部结构分析准确率作为评价指标之一，当一个词被正确识别而且其所有内部依存弧都正确分析时，我们才认为该词的内部结构被正确分析（与短语结构句法分析中的内部结构评价完全等价）。

## 2.6.2 基于字的短语结构句法分析

### 2.6.2.1 开发数据集上的结果




a) 分词词性标注F值.　　b) 句法结构F值.

图 2-12 分词词性标注F值以及句法结构F值与柱大小以及训练迭代次数的关系。

Fig. 2-12 Accuracies for joint segmentation and tagging as well as constituent parsing using beam sizes 1, 4, 16 and 64, respectively.

图2-12显示了在感知器训练过程中随着训练次数的增加以及柱大小的增加这种基于字的句法分析模型在开发数据集上的测试性能变化。从图中我们可以看到，基于字的句法分析模型随着柱大小的增加，其性能也在逐渐增加，但是当柱大小增加到一定程度，性能的增加程度也在逐步降低；与此同时，模型的速度在逐步下降，我们在Intel Core i5-3470 CPU (3.20GHz)，Fedora 17 以及gcc 4.7.2环境下运行该模型，柱大小为1，4，16，64时相应的解码速度为318.2句每秒，98句每秒，30.3句每秒以及7.9句每秒。根据这个实验结果，经过综合考虑后，我们采用64柱大小来进行后面的实验。

进一步，我们还在开发数据集上测试了一下词结构带来的字特征（如

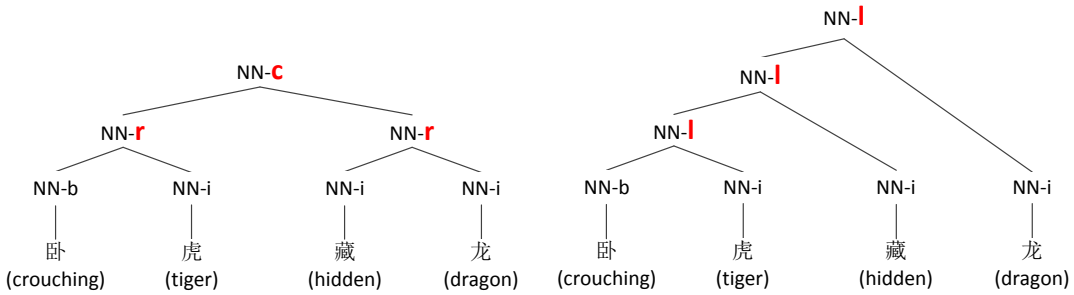
表2-1所示粗体部分）的效用，实验结果如表2-5所示，从实验结果中我们可以看到，这类特征在基于字的联合模型上效果比较明显。

表 2-5 词结构引入的字特征的效用测试.

Table 2-5 Results of feature ablation for word structure features.

| | 分词 | 词性 | 句法 | 词结构 |
|---|---|---|---|---|
| 包含词结构特征 | 96.76 | 94.17 | 85.34 | 96.37 |
| 不包含词结构特征 | 96.53 | 93.74 | 84.75 | 95.92 |

### 2.6.2.2 测试数据上的结果




a) 人工标注词结构　　b) 自动词结构

图 2-13 人工标注词结构和自动伪词结构的一个比较例子。

Fig. 2-13 Examples for annotated and pseudo word structures

为了验证这种基于字的句法分析的优势，我们设计了两个基准系统，第一个系统是一个串行模型，串行系统先对句子进行分词和词性标注，使用张岳和Clark在2010年提出的分词词性标注联合模型，然后在这个基础上对该句子进行句法分析，使用张岳和Clark在2009年提出的基于转移的短语结构句法分析器，通过和这一个基准系统进行比较，我们来表明基于字的一体化句法分析相对于串行模型的优势；第二个系统在解码以及训练算法上和我们提出的基于字的系统完全一样，但是使用自动的词结构，如图2-13所示给出了人工标注词结构和自动词结构的一个比较例子，这个系统也是由我们首次提出，设计这个基准系统是为了验证真实的汉语词结构对句法分析具有促进作用。表2-6显示了我们最终的结果，其中的数字表明，一体化的系统无论在采用人工标注的真实词结构还是自动词结构都会比串行模型的性能要好，采用人工标注的真实词结构会更进一步提升句法的性能。

表 2-6 测试集上的最终结果。

Table 2-6 Final results on test set.

| | 任务 | P | R | F |
|---|---|---|---|---|
| 串行模型 | 分词 | 97.35 | 98.02 | 97.69 |
| | 词性 | 93.51 | 94.15 | 93.83 |
| | 句法 | 81.58 | 82.95 | 82.26 |
| 伪词结构 | 分词 | 97.32 | 98.13 | 97.73 |
| | 词性 | 94.09 | 94.88 | 94.48 |
| | 句法 | 83.39 | 83.84 | 83.61 |
| 人工标注词结构 | 分词 | 97.49 | 98.18 | **97.84** |
| | 词性 | 94.46 | 95.14 | **94.80** |
| | 句法 | 84.42 | 84.43 | **84.43** |
| | 词结构 | 97.22 | 97.94 | 97.59 |

表 2-7 和前人工作的比较。

Table 2-7 Comparison with previous work.

| 系统 | 分词 | 词性 | 句法 |
|---|---|---|---|
| Kruengkrai+ ’09 | 97.87 | 93.67 | – |
| Sun ’11 | 98.17* | 94.02* | – |
| Wang+ ’11 | 98.11* | 94.18* | – |
| Li ’11 | 97.3 | 93.5 | 79.7 |
| Li+ ’12 | 97.50 | 93.31 | – |
| Hatori+ ’12 | **98.26*** | 94.64* | – |
| Qian+ ’12 | 97.96 | 93.81 | 82.85 |
| 我们的串行模型 | 97.69 | 93.83 | 82.26 |
| 我们的模型使用伪词结构 | 97.73 | 94.48 | 83.61 |
| 我们的模型使用人工标注的词结构 | 97.84 | **94.80** | **84.43** |

#### 2.6.2.3 前人工作的比较

进一步，我们和其它相关工作进行了比较，比较结果如表2-7所示，其中“*”表示额外的资源在他们的模型中被用到，Kruengkrai+ ’09 表示Canasai Kruengkrai等人在2009年提出的基于网格错误驱动的分词词性标注联合模型[76]；Sun ’11 表示由孙薇薇提出的一个基于子词的分词词性标注联合模型[60]，这一系统使用了习语词典；Wang+ ’11 表示由Yiou Wang等人提出的一个半指导的串行分词词性标注模型[77]，其中用了大规模的未标注数据；Li ’11 表示由李中国在2011年提出的一个基于词缀词结构的分词、词性标注以及短语句法的生成模型[67]；Li+ ’12 表示由李中国和周国栋在2012年提出的一个一体化的分词词性依存句法联合模型[70]，这个模型也建立在基于词缀的词结构基础之上；Hatori+ ’12 表示由Jun Hatori等人在2012年提出的一个分词词性标注和依存句法的联合模型[69]，他们使用了Hownet以及中文维基百科这两种外部词典；Qian+ ’12 表示有Xian Qian和Yang Liu在2012年提出的一个分词、词性标注以及短语句法的联合模型[68]，他们在训练时分别使用了各自的模型训练方法，而在解码时，将这三种模型联合在一起。从整体上看，我们这种基于字的句法分词在词性标注和句法分析上取得了最好的性能。

### 2.6.3 基于字的依存结构句法分析

#### 2.6.3.1 基准模型和我们新提出的模型

我们所使用的基准模型是串行模型，它由两部分组成，第一部分是一个联合的分词词性标注模型，其具体实现方法如Zhang and Clark (2010)年的文章所示[3]，第二部分是一个基于词的依存分析模型，它既可以是黄亮等人(2009)提出的标注弧转移柱搜索算法[39]，也可以是Zhang and Clark (2010）年提出的贪心弧转移柱搜索算法[38]，因此，综上所述我们有两种基准模型，分别命名为STD (pipe)（标准弧转移算法）和EAG (pipe)（贪心弧转移算法）。对于这两个模型，第一部分分词词性标注的联合模型我们所使用的柱大小为16，第二部分依存结构句法分析的模型我们所使用的柱大小为64，这些设置都对应于上述模型达到最佳效果时所使用的设置。

上面所说的是基准模型，我们同时还提出了六种新的模型，它们分别为：

- STD (real, pseudo)：标准弧转移算法解码，使用人工标注的词内部结构和伪造的词之间的依存，换句话说，它没有分析出词之间的依存结构，只是得到了分词、词性标注以及词内部结构的结果；

- STD (pseudo, real)：标准弧转移算法解码，使用伪造的词内部结构和人工标注的词之间的依存，它没有分析出词的内部结构，实际上它就是一个简单的分词、词性标注以及依存句法联合模型；
- STD (real, real)：标准弧转移算法解码，使用人工标注的词内部结构和人工标注的词之间的依存，它能同时分析出分词、词性标注、依存句法以及词内部结构；
- EAG (real, pseudo)：贪心弧转移算法解码，使用人工标注的词内部结构和伪造的词之间的依存，换句话说，它没有分析出词之间的依存结构，只是得到了分词、词性标注以及词内部结构的结果；
- EAG (pseudo, real)：贪心弧转移算法解码，使用伪造的词内部结构和人工标注的词之间的依存，它没有分析出词的内部结构，实际上它就是一个简单的分词、词性标注以及依存句法联合模型；
- EAG (real, real)：贪心弧转移算法解码，使用人工标注的词内部结构和人工标注的词之间的依存，它能同时分析出分词、词性标注、依存句法以及词内部结构。

其中人工标注的词内部结构是指内部依存结构，从词短语内部结构直接转化而来，而伪造的词内部结构是$c_1 \curvearrowright c_2 \curvearrowright \cdots \curvearrowright c_m$，其中$c_i$为一个词中的第i个字；人工标注的词之间的依存结构是指从短语结构句法树中根据父亲节点发现规则自动转换出来的依存树，而伪造的词之间的依存结构是$w_1 \curvearrowright w_2 \curvearrowright \cdots \curvearrowright w_n$，其中$w_i$为一个句子中的第i个词。所有的这些模型我们所使用的柱大小都是64，这是我们根据前面实验的经验所设置的，考虑到了速度和性能双方面的影响。

#### 2.6.3.2 测试数据上的最终结果

表2-8中显示了上面几个模型在CTB 5.0测试集上的最终结果；同时我们将我们提出的方法和Hatori在2012年提出的一个分词、词性标注和依存句法联合模型做了一个比较。首先我们将STD (pipe)和STD (real, pseudo)相比较，以及将EAG (pipe)和EAG (real, pseudo)相比较，STD/EAG(pipe)模型中的分词和词性标注性能和STD/EAG(real, pseudo)反应了词内部结构给分词、词性标注带来的变化，我们可以看出，词内部结构可以带来更高性能的词法分析。进一步将STD (pipe)和STD (pseudo, real)相比较，以及将EAG (pipe)和EAG (pseudo, real)相比较，这两个比较能表明联合模型在词法分析和句法分析上的优势，因为STD/EAG(pipe)是一个串行模型，而STD/EAG(pseudo, real) 是没有使用词内部结构时的一个联合模型。Hatori+ ’12 也是一个联合模型，但是我们提出的联合模型能获取更高性能的准确率，这是由于我们所采用的转移系统和他们的不一

样。最后我们将STD (pseudo, real)和STD (real, real)相比较，以及将EAG (pseudo, real)和EAG (real, real) 相比较，这两个比较表明了词内部结构在句法分析上的有效性，表上的结果也反应了词内部结构确实能提高依存结构句法分析的性能。

表 2-8 测试集上的最终结果。

Table 2-8 Final results on test set.

| | 分词 | 词性 | 句法 | 词结构 |
|---|---|---|---|---|
| STD (pipe) | 97.69 | 93.83 | 80.28 | – |
| STD (real, pseudo) | **97.95** | 94.05 | – | **97.60** |
| STD (pseudo, real) | 97.87 | 94.28 | 81.63 | – |
| STD (real, real) | 97.84 | **94.62** | **82.14** | 97.30 |
| Hatori+’12 | 97.75 | 94.33 | 81.56 | – |
| EAG (pipe) | 97.69 | 93.83 | 80.29 | – |
| EAG (real, pseudo) | **97.90** | 94.11 | – | **97.65** |
| EAG (pseudo, real) | 97.76 | **94.36** | 81.70 | – |
| EAG (real, real) | 97.84 | **94.36** | **82.07** | 97.49 |

## 2.7 本章小结

本章提出了使用字作为中文句法分析的基本单位，在此基础上提出了基于字的中文短语结构句法分析和基于字的中文依存结构句法分析。我们为基于字的中文短语结构句法分析提出了一种基于转移的模型，而为基于字的中文依存结构句法分析提出了两种不同的转移模型，它们都分别从基于词的转移算法中进行扩展而得到。最终的实验结果也表明了这几种基于字的联合模型确实能有效提升分词、词性标注以及句法分析的性能。

词内部结构的分析是过去中文处理中被忽略的一个任务，而实际上中文词确实是有内部结构的，我们通过一些例子表明了这一观点，同时采用短语结构的表现形式标注了部分中文词的内部结构。我们最终提出的基于字的句法分析方法就是通过词内部结构而形成的。实验表明，在引入词内部结构后，分词、词性标注以及句法分析的性能都比又能比纯粹的联合模型有所提升。

# 第 3 章 句法依存语义依存联合模型研究

## 3.1 引言

语义分析是自然语言处理句子级研究的终极目标，但是关于语义分析的研究方法，不同的研究者有着不同的观点，总体上来分两大派别，其中一种更注重于词汇语义学(Lexical Semantics)，而另外一种更倾向于组合语义学(Compositional Semantics)。基于词汇语义学的研究方法通常面向单独的词建立一个语义知识库，例如针对英文的Wordnet[43]。针对中文这样的资源比较多，具有代表性的工作包括知网(Hownet)，同义词词林以及同义词词林扩展版[44, 45, 78]。最近，不少研究者通过深度学习来自动获取词的语义表示[79, 80]，其基本原理是利用一个词的上下文去刻画它的含义，最终每个词的语义表示是通过一个多维的实数向量来体现的，词与词之间的语义联系通过相应的向量运算来得到，例如“VEC(王后)-VEC(女人)=VEC(国王)-VEC(男人)”。基于词汇语义学的研究方法也可以是面向指定的上下文来分析一个词的具体含义，句子级的词义消歧便是一个典型的基于词汇语义任务[46, 47]，它针对一个句子中的特定词，对该词所表示的真实含义进行区分。例如“近期他要打算入手一个**苹果**手机”，其中的“苹果”就是一个含有歧义的词，这个词在语义知识库中一般会包含有“与吃水果相关的苹果”和“公司名字”两个义项，而词义消歧便是区分在该句子中“苹果”属于哪个义项。

第二种观点是采用组合语义学的观点来进行研究，这种研究尝试把一个句子中的所有词语，通过逻辑组合的方式进行逐步的合并，最后用一个类逻辑的语法将句子的语义表示出来，并能为机器所处理。这一类研究观点的初期工作由John M. Zelle 和Raymond J. Mooney 在1996年提出，他们将自然语言的句子转换成数据库中的SQL查询语句[81]。具体的分析方法有基于规则的方法[82]，基于机器翻译的方法等等[83]。进一步研究者建议使用一种更为高级的逻辑表达式Lambda calculus来表示句子的逻辑语义[48]，最早的工作包括Zettlemoyer和Collins提出的使用组合范畴文法作为中间句法来将一个句子转换成Lambda表达式[84]，在其基础上有不少的后续的跟进工作[49, 50, 85]。

基于组合语义的研究方法能比词汇语义的方法分析得更深入，但是这种方法存在着两个方面的不足。第一个问题是领域受限，过去这种研究工作所使用

的训练和测试语料的句子总数一般不超过1,000，其最主要的原因在于组合语义的复杂表示结构，从而语料的构建很难扩展到大规模通用领域的句子。第二个问题是词汇的抽象比较难，这个问题在词义消歧中也同样存在。针对这两个问题，我们采用面向语义的依存分析（语义依存分析）来作为一个过渡工作，这种表示方法存在着两个方面的优点。第一个优点是语料相对来说比纯粹的面向逻辑的语料容易构建，目前一些相关的语料已经存在[55, 86]。第二个优点是我们并不需要去抽象词汇本身，而是去通过该词汇所能承受的语义框架来描述该词汇，因为这些论元的数目相对于词汇来说是非常有限的。根据不同的语义依存标注规范，这些论元的数目略有不同，目前论元数目最多的语义依存语料中包含有大约120多种语义关系。

关于语义依存的分析方法，大部分研究者仍然沿用着和句法依存分析完全相似的算法，仅仅认为语义依存分析的语料发生了变化。事实上，语义依存分析的层次要高于句法依存分析，但是却很少有研究者关注句法信息对语义依存分析的帮助作用。过去的工作一般将面向语义的依存分析和面向句法的依存分析看作两个单独的平行任务，认为它们仅仅是两种不同的依存分析，而我们将在本章中表明，句法信息也可以非常有效的帮助语义依存分析。进一步基于这一结论，我们将提出一种联合模型的方法，使得这两种依存分析能在同一个统计模型中同时完成，并且这两种依存分析的性能都能得到一定的提升。

在本章中，我们的主要目的是提出一种语义依存分析和句法依存分析的联合模型。首先我们对语义依存分析做一个简要的介绍；然后通过另一种被广泛关注的浅层语义分析手段——语义角色标注(Semantic Role Labeling, SRL)[52]——来对语义依存分析的合理性进行研究，以表明语义依存分析作为一种语义分析手段的合理性；进一步我们对中文语义依存和句法依存做一下对比分析，并表明句法分析能进一步促进语义依存分析的性能，这样为句法和语义依存分析的联合模型提供潜在的证据；最后我们提出一个联合模型，来同时处理语义依存分析和句法依存分析，并且最终的结果也表明了联合模型的有效性。

## 3.2 相关工作

在英文上面，使用依存文法来表示语义结构的工作主要包括Johansson等人的工作[87]，以及后来的Stanford依存结构[88]。基于依存文法的中文语义依存表示最早由李明琴等人在2003年提出[55]，他们对知网的一些语义关系做了一定筛选，并且在句法和语义之间做了一些协调，产生了一个面向语义的依存标注规

范，在此基础上建立了一百万词规模的标注语料。类似，Jiajun Yan以及王丽杰等人也利用了相似的语义关系，对中文宾州句法树库中的短句子也做了语义标注，但是Jiajun Yan所标注的语料所使用的句子都是短句[56]，而王丽杰等人的标注工作包含任意长度的句子[86]，并且王丽杰等人标注的这部分语料，在经过进一步的整理之后，由车万翔等人在SemEval-2012上被用来组织了国际公开评测，国内不少研究机构都参与了这一评测[57]。斯坦福大学自然语言处理组也提出了一套面向语义的依存规范，即中文Stanford依存，但是由于中文Stanford依存里面是通过句法树库直接转化得到的，所以仍然存在着大量的句法信息[89]。

过去在验证语义依存的实用性方面的工作，主要都是在英文语义依存分析的基础上进行展开的。一般所采取的方法是选取一些特定的自然语言处理的应用，例如机器翻译、情感识别，语义角色标注等等，观察语义依存和句法依存在这些任务上的效果对比[90]。在中文语义依存方面，除了张碧娟等人使用机器翻译去验证了中文Stanford依存的有效性之外[89]，没有其它的研究工作。

对于语义依存的分析方法，大部分的工作都是套用现有的句法依存分析的方法来直接进行语义依存分析。例如基于图的依存分析方法[91–93]，基于转移的依存分析方法等等[94]。这些分析方法甚至都完全沿用以往句法依存分析所使用那些特征，因此这些方法都忽略了句法信息对语义依存分析的潜在帮助。本章我们的工作将表明，句法信息对于提高语义依存分析性能仍然有用。进一步，我们提出一种句法依存分析和语义依存分析的联合模型，使得这两种依存分析能在同一个模型中同时进行，这样能避免先进行句法依存分析后进行语义依存分析这种串行模式所带来的错误蔓延问题。

## 3.3 中文语义依存表示

中文语义依存是建立在依存文法基础之上，它将句子的语义表示成为一个有向树，树上的边称为依存弧，弧上具有特定的关系$l$。每条弧将句子中的两个词$w_i$和$w_j$进行关联($w_i \overset{l}{\frown} w_j$)，其中$w_i$称为弧的核心节点或者父亲节点，而$w_j$称为孩子节点；弧上的关系$l$表示这两个词之间的语义关系。图3-2中的上面部分显示了一棵在SemEval-2012评测中所使用的语义依存树，我们称之为HIT语义依存树。具体来讲，最后的这棵树满足以下四个条件：

(1) 每个句子中有且仅有一个词为该句子的核心（该词的父亲节点不在句子内部），或者说该词为这棵有向树的根节点；

(2) 语义依存树必须是弱连通的，树中的任何一个非根节点(词)都必须有

一个父亲节点，后者是前者的充分条件；

(3) 树中的任何一个非根节点(词)都只能有一个父亲节点；

(4) 如果将树按句子中的词序进行展开平铺，那么这棵树不存在交叉弧。

HIT语义依存树是根据中文宾州句法树库利用一定的短语句法到语义依存的转换规则进行转换而得到的，并且在此转换基础上还进行了多次人工校验。HIT语义依存树的依存弧建立是以语义为导向的，对于一些没有实际意义的功能性的词语，在语义依存树中不再作为任何词的核心，而是作为叶子节点。对于弧上关系，HIT语义依存关系是在HowNet语义体系为基础上精心设计的[44]，并综合考虑了其它各家语义体系，包括鲁川先生提出的意合网络[95]、袁毓林先生提出的语义关系标注体系[96]、冯志伟先生根据依存语法提出的中文论元框架以及林杏光先生提出的二十二个基本格[97, 98]，最终的语义依存关系如表3-1所示。表中的语义关系和前人提出的语义依存关系相比，增加了反语义关系，谓语省略语义关系以及名词谓语语义关系，同时语义依存关系中还保留了部分具有句法色彩的标签，例如并列，让步，顺承等等。

语义依存分析可以为基于组合语义的分析方法提供一定的支持。对于图3-2中的语义依存例子，我们可以发现，“提出”的孩子节点共有三个，分别为“建议”、“了”和“措施”，它们各自的语义关系分别为“experiencer（经验者）”、“aspect（体，表示完成时态）”和“content（内容）”。这样，我们便可以得到三个独立的谓词结构：“提出（experiencer：建议）”、“提出（aspect：了）”和“提出（content：措施）”。如果我们还能知道“提出”的两个必要论元角色是experiencer和content时，我们便可以对这两个论元进行组合合并得到“提出（experiencer：建议，content：措施）”。这一步合并需要词汇语义知识库的支持，如果有了这一步，我们便可以非常容易的得到整个句子逻辑语义表示。

## 3.4 和句法依存分析的对比

语义依存分析和句法依存分析的核心文法都是依存文法，从表面上来看没有任何区别，但是从本质上看，由于它们的分析目的不同导致了它们建立依存结构的规范存在着比较大的区别。首先我们从依存弧建立的原则来看，图3-1显示了语义和句法之间巨大差别的一个例子。对于句法依存分析，“的”字是“美国”与“华人”之间的一个中间载体，而对于语义依存分析，“美国”与“华人”之间进行直接的关联。

其次在具体的依存关系上面，我们采用了如表3-1所示的语义标签，这和句

表 3-1 HIT语义依存关系一览表。

Table 3-1 The HIT semantic dependency labels.

| | |
|---|---|
| **主语义角色** | |
| 主体语义角色 | 施事\|agent，经验者\|experiencer，致事\|causer，领有者\|possessor，存现体\|existent，整体\|whole，关系主体\|relevant |
| 客体语义角色 | 类指\|isa，内容\|content，占有物\|possession，受事\|patient，部分\|OfPart，损益者\|beneficiary，参照体\|contrast，相伴体\|partner，依据\|basis，原因\|cause，代价\|cost，范围\|scope，关于\|concerning |
| **辅助语义角色** | |
| 时间类 | 时段\|duration，终止时间\|TimeFin，起始时间\|TimeIni，时间点\|time，时间状语\|TimeAdv |
| 空间和状态类 | 终处所\|LocationFin，原处所\|LocationIni，通过处所\|LocationThru，终状态\|StateFin，状态\|state，原状态\|StateIni，方向\|direction，距离\|distance，处所\|location |
| 连动，方式，状态类 | 伴随\|accompaniment，接续\|succeeding，泛指频率\|frequency，工具\|instrument，材料\|material，手段\|means，角度\|angle，动量\|times，顺序数\|sequence，顺序量\|sequence-p，否定\|negation，程度\|degree，情态\|modal，强调\|emphasis，方式\|manner，体\|aspect，插入语\|comment |
| **定语语义角色** | |
| 直接修饰类 | 领有者\|d-genetive，类别\|d-category，成员\|d-member，事域\|d-domain，名量\|d-quantity-p，数字\|d-quantity，指量\|d-deno-p，指示\|d-deno，宿主\|d-host，时间短语修饰语\|d-TimePhrase，地点短语修饰语\|d-LocPhrase，机构短语修饰语\|d-InstPhrase，属性\|d-attribute，限定\|d-restrictive，材料\|d-material，内容\|d-content，顺序数\|d-sequence，顺序量\|d-sequence-p，未知\|qp-mod |
| 动词修饰名词(VP短语) | r-主语义角色，如：反施事\|r-agent，反受事\|r-patient，反领属者\|r-possessor (不定，任何主语义角色都可以) |
| 名词修饰名词(省略谓语) | c-主语义角色，如：隐施事\|c-agent，隐内容\|c-content，隐受事\|c-patient (不定，现有语料中出现了4种) |
| 名词谓语(NP短语) | j-主语义角色，如：间接施事\|j-agent，间接受事\|j-patient，间接对象\|j-target (不定，任何主语义角色都可以) |
| **句法语义角色以及其它** | |
| 句法语义角色 | 原因\|s-cause，让步\|s-concession，假设\|s-condition，并列\|s-coordinate，选择\|s-or，递进\|s-progression，除外\|s-besides，顺承\|s-succession，目的\|s-purpose，措施\|s-measure，割舍\|s-abandonment，选取\|s-preference，总括\|s-summary，分述\|s-recount，关于\|s-concerning，结果\|s-result |
| 其它类别 | 助词-方位词-连词\|aux-depend，介词\|prep-depend，标点符号\|PU，根\|ROOT |

法依存分析通常所采用的主语、谓语、宾语、定语、状语以及补语等等有着显著的区别，语义的标签相对于句法来说更丰富，能体现更多的语义上的细微差别。

进一步我们通过一个具体的例子将HIT语义依存树和一个句法依存树进行更为直观的比较，这棵句法依存树是通过张岳和Clark在2008年提出的父亲节点发现规则将中文宾州短语句法树库进行自动转换而得来的（用MALT表示）[38]。图3-2显示了它们之间的一个直接对比，即针对一个句子我们同时给出了它的HIT语义依存树和MALT句法依存树，其中上面部分HIT语义依存树，下面部分是MALT句法依存树。

从图中我们可以看出，语义依存树很大程度上忽略了一些虚词，实词与实词之间更倾向于直接建立联系，例如“在”和“的”等词都不再像句法依存树那样作为沟通两个实词的桥梁。表3-2对这两种依存从典型的依存关系、关系数目以及标注难度上做了对比。从表中可以看出，语义依存的标签数目要比句法依存多出了十倍左右，同样其标注难度也比句法依存高出了很多，这一点导致了实际的语义依存分析语料的规模要远远小于句法依存分析的语料规模，这也是语义依存分析所面临的一个大困难。

前面的比较都基于标注规范而进行的，我们将进一步从实际数据出发来对语义依存分析和句法依存分析进行比较。我们使用同样的句子集合作为训练和开发语料，同时也采用同样的依存分析算法来训练最终的依存分析模型，观察它们在同样句子集合的测试集上的性能对比。从理论上来说，一方面弧上标记关系越多则自动依存分析难度越大，这一点是显然的，因为标签数目的增加会增加数据的稀疏性；另一方面如果长距离依存弧越多则自动依存分析难度越大，这一点已经在前人的分析工作中得到了体现[99]。对于依存关系数目的对比表明了语义依存分析的难度要高于句法依存分析。对于长距离依存弧的对比，我们也观察实际语料，发现在HIT和MALT标注规范中，依存弧平均


a) 句法依存　　b) 语义依存

图 3-1 语义依存和句法依存的一个对比。

Fig. 3-1 Comparison between semantic and syntactic dependencies.

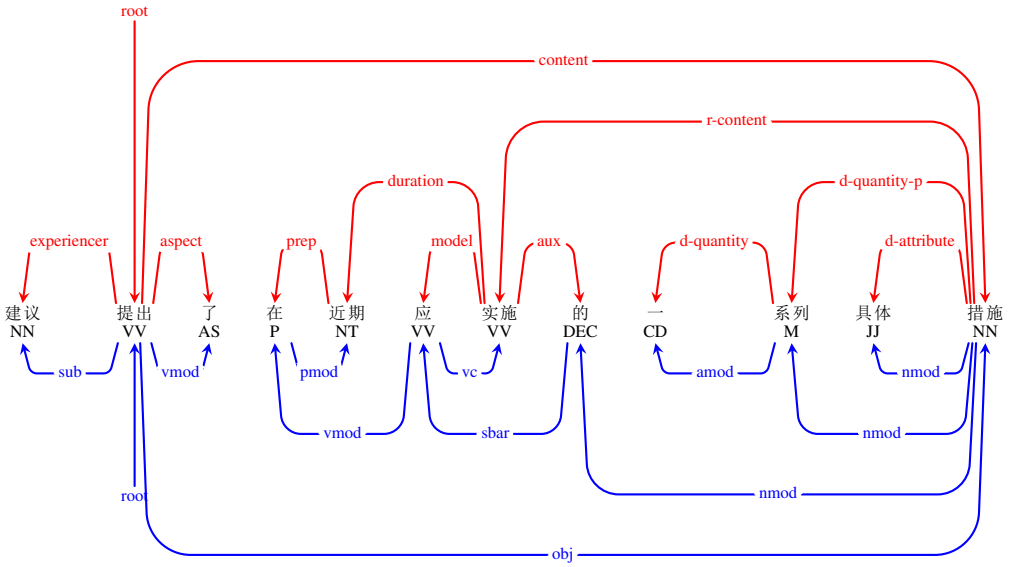



图 3-2 HIT语义依存和MALT句法依存的一个对比例子。

Fig. 3-2 An example to compare the HIT and MALT dependencies.

表 3-2 HIT语义依存和MALT句法依存关系对比以及标注难度上的对比。

Table 3-2 A comparison between the HIT and MALT dependency labels.

| 依存分析种类 | 典型的依存关系 | 依存关系数目 | 标注代价 |
|---|---|---|---|
| HIT | agent,patient,content | 122 | ★ ★ ★ |
| MALT | sub, obj, vmod, nmod | 12 | ★ |

距离分别为3.05和2.87，同时图3-3也表明了HIT语义依存分析的长距离依存要多于MALT句法依存分析。因此这一比较也表明了语义依存分析的难度要高于句法依存分析。我们将通过具体的实验来验证这一观点，具体情况将在后面的实验部分介绍。

## 3.5 中文语义依存分析合理性研究

虽然语义依存分析已经被提出了很长时间，但是很少有后续的研究工作尝试于改进语义依存分析的性能，其原因有两点，其中第一点是语义依存分析的相关方法大部分和传统的依存分析类似，另外一点比较重要的是语义依存分析的合理性没有得到比较好的证据支持，即为什么要研究语义依存，一方面和其它语义分析方法相比，它能否更丰富的表达语义，另一方面和其它依存规范相比，其优势到底在哪里?

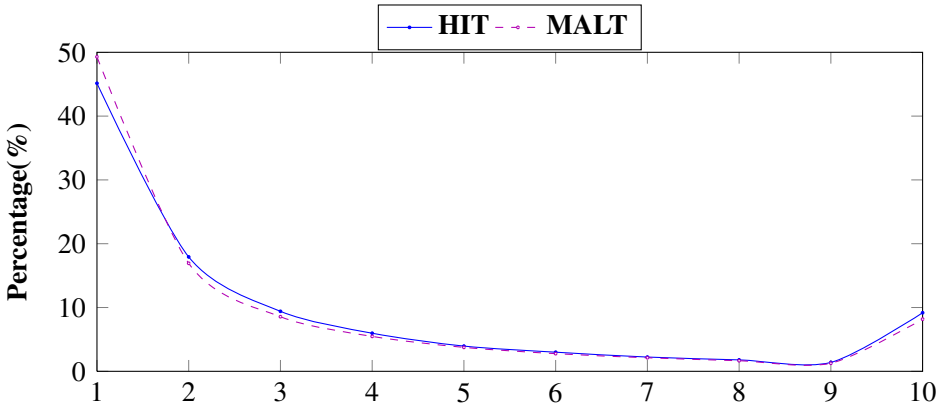



图 3-3 HIT和MALT依存弧距离分布情况。

Fig. 3-3 Distance distribution of HIT and MALT dependencies.

在本章中我们将尝试解决语义依存的合理性问题，我们将通过另外一种被广泛认可的语义分析手段——语义角色标注(Semantic Role Labeling, SRL)——作为媒介，来回答上面提出的关于语义依存合理性的两个问题。对于第一个问题，我们直接对比HIT语义依存和语义角色标注的异同，来表明HIT语义依存具有更丰富的语义信息。而对于第二个问题，我们将语义角色标注作为依存分析的高层应用，比较HIT语义依存和MALT句法依存在语义角色标注上的性能，便可以知道语义还是句法依存标注哪种规范更适合于语义角色标注。

### 3.5.1 语义角色标注任务介绍

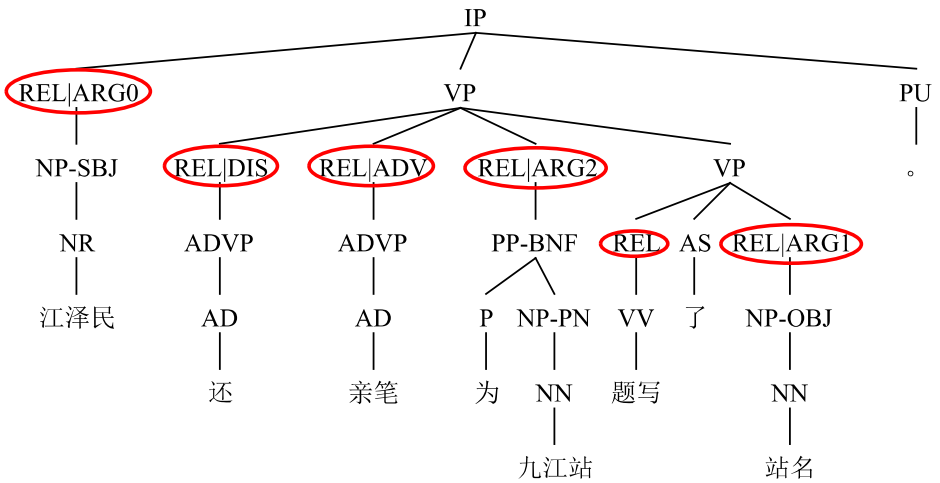



图 3-4 语义角色标注的一个例子。

Fig. 3-4 An example of Chinese semantic role labeling.

语义角色标注是目前普遍采用的浅层语义分析手段之一，它的目的是为句子中的谓词去寻找论元参数，同时指定每个论元的内容。一个语义角色表示一个谓词和句子中其它词语的语义关系，这个关系可以是施事(agent)，受事(patient)，时间(time)，地点(location) 等等，谓词既可以是名词也可以是动词。在实际的中文语义角色标注中，语义角色的关系一般被抽象了，目前主要使用了6个主论元结构，用ARG0…ARG6表示，以及14 个功能参数论元，如表3-3所示。图3-4显示了语义角色标注的一个例子，其中Rel表示谓词，而其它被椭圆标记的部分指明了这个谓词的论元以及论元的内容。和语义依存树的表示方法类似，语义角色标注也是词汇语义与组合逻辑语义之间的一个较好的平衡点，首先针对具体谓词的分析体现了该谓词的词汇语义，而针对谓词，分析其语义框架以及各个框架的成分，便使得该动词的完整逻辑组合语义体现了出来。

| 标签 | 描述 | 标签 | 描述 |
|---|---|---|---|
| ADV | 副词 | FRQ | 频率 |
| BNF | 受益者 | LOC | 地点 |
| CND | 条件状语 | MNR | 方式 |
| DIR | 方向 | NEG | 否定 |
| DIS | 篇章连接词 | PRP | 目的或者原因 |
| DRG | 程度 | TMP | 时间 |
| EXT | 范围 | TPC | 主题 |

表 3-3 语义角色标注中的功能参数论元。

Table 3-3 The function labels of semantic role labeling.

### 3.5.2 语义依存分析和语义角色标注对比

语义依存分析是对整个句子进行语义分析，其目标在于深度语义分析；而语义角色标注是针对句子中的核心谓词进行语义分析，主要包括部分动词和名词，实际上是一种浅层的语义分析。在这里我们简单的归纳一下语义依存分析和语义角色标注的区别，主要从四个方面进行描述：

(1) 从表示结构上面来看，语义依存分析最终得到一棵完整的语义依存树，而语义角色标注得到的是谓词以及和这个谓词相关的语义角色成分。

(2) 从关注对象来看，语义依存分析关注整个句子，而语义角色标注是针对独立的谓词进行分析；语义依存分析的标注对象包含语义角色标注的标注对

象。

(3) 从标注关系内容来看，语义依存蕴含有更丰富的语义信息。对于核心语义关系，语义角色标注只能简单的区分为六类，而语义依存分析有几十类；对于其他语义关系，语义依存分析的区分也是更为细致。

(4) 从分析结果的关联性与完整性来看，语义依存分析强调一个整体，而语义角色标注强调独立的谓词个体。

通过上述的对比分析，我们可以看出语义依存分析能更全面的分析一个中文句子所蕴含的语义，不过由于依存文法形式约束过多，语义依存分析也存在着一定的缺陷有待于以后进一步的改进，但是从总体上而言，语义依存分析做为一个语义分析的手段是比较合理的。

### 3.5.3 基于依存的语义角色标注系统

语义角色标注是一种浅层的语义分析任务，自从2003年其就有很多相关的研究工作[51, 52, 100–105]，它作为一种语义分析手段被得到广泛的认可。为了验证语义依存分析的有效性，我们构建一个基于依存分析的语义角色标注系统，然后分别使用语义依存分析和句法依存分析的结果，比较语义角色标注受到的影响。如果语义依存分析的结果能更好的帮助语义角色标注，则表明语义依存分析和语义角色标注更为贴近，从而表明了语义依存分析的有效性。为了避免自动依存分析的难度所造成的比较的不公平性，我们在比较过程中都假定正确的依存树已经给定。

我们采用一种基于序列标注的方法来进行语义角色标注分析。首先，对于任何一个谓词，我们使用块边界类别区分的方法，将一个语义角色块转换成为一组标注序列，如图3-5所示。对于谓词，我们用“Rel”表示；对于(Temporal, TMP)“在 近期”，我们用“B-TMP”和“I-TMP”表示这两个词的标签；而对于“一 系列 具体 措施”，除了“一”被标记为“B-A1”之外，其它词都被标记为“I-A1”；不属于这两种语义角色的其它非谓语词，我们用“O”表示。对于“BIO”的三个字母的具体含义，我们解释如下：“B”表示一个语义角色块的开始，“I”表示一个语义角色块的继续，“O”表示不属于任何语义角色。值得注意的是，有些特殊情况下一个语义角色块可能会跨越多个连续的短语块，这种情况下后续的短语块的第一个词用“C-XX”标记。

在这个基于依存分析的语义角色标注系统中，我们所抽取的特征和车万翔等人2009年参加CONLL09句法和语义评测任务的系统中所使用的特征完全一

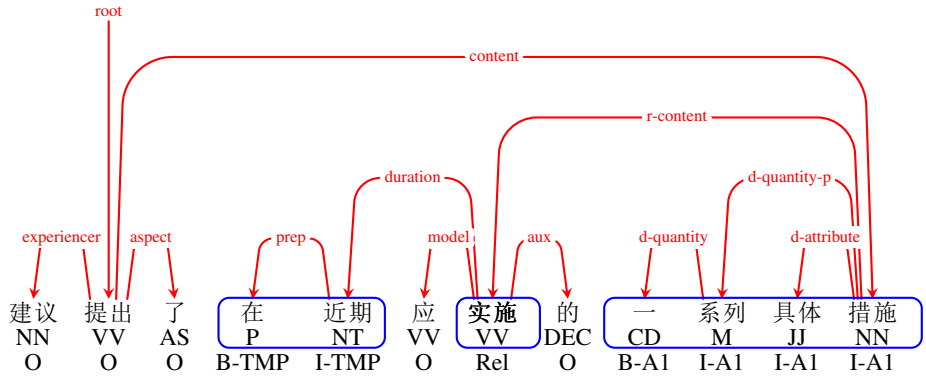



图 3-5 基于序列标注的语义角色标注示意图。

Fig. 3-5 Semantic role labeling systems based on sequence labeling.

样[106]。在训练这些特征的权重时，我们采用了条件随机场(Conditional Random Field, CRF)[19]。在进行最终的解码，我们分为两步进行，首先我们使用CRF模型结合前向后向算法输出每个词所有可能类别的边缘概率，然后在这个边缘概率的基础上，我们利用整数线性规划进行解码。假设所有可能的输出类别构成的集合为$L = l_1, \cdots, l_{\|L\|}$，待解码的句子所含有的词的数目为$n$，则我们一共定义了$n \| L \|$个变量，分别假设为$a_{ij}$，其中$0 < i \le n$而且$0 < j \le \| L \|$，这些变量都只能取0或者1，如果$a_{ij} = 1$，则表示第$i$个词的最终标记为$l_j$。解码时，整数线性规划的目标是使得整个句子的所有词语和它们最终的标签的联合概率达到最大，即使得$\sum_{0<i\le n, 0<j\le\|L\|} a_{ij} \cdot \log p_{ij}$最大，其中$p_{ij}$就是CRF模型输出的边缘概率，其中相关的标签约束包括以下五类：

(1) 每个词只能有一个标签；

(2) I-XX标签前面必须是I-XX、B-XX或者C-XX；

(3) C-XX前面至少有一个B-XX；

(4) B-XX前面不能是I-XX。

(5) 核心语义角色标签A0~A5不能重复出现。

## 3.6 句法依存对语义依存影响

本章的最终目的在于建立一个语义依存分析和句法依存分析的联合模型。前面的工作我们主要是为了表明中文语义依存分析作为语义分析手段的合理性，从本节开始，我们将逐步提出我们最终的联合模型。首先在本节我们将讨论这样一个问题，中文句法依存分析能从多大程度上帮助中文语义依存分析?

只有当中文句法依存分析能够有效的帮助中文语义依存分析时，语义依存分析和句法依存分析的联合模型才有可能发挥作用。我们所使用的句法依存分析为MALT依存分析，语义依存分析为HIT依存分析。

### 3.6.1 语义依存和句法依存对应关系

表 3-4 句法与语义依存弧一致性关系。

Table 3-4 Head comparisons between semantic and syntactic dependencies.

| 词性对(前五) | 一致性(%) | 词性对(后五) | 一致性(%) |
|---|---|---|---|
| DT◠M | 99.85 | DEG◠NN | 0.05 |
| AS◠VV | 99.67 | P◠NN | 0.08 |
| CD◠M | 99.35 | LC◠NN | 0.17 |
| CC◠NN | 97.70 | DEC◠VV | 0.17 |
| DT◠NN | 97.35 | DEC◠VA | 0.23 |
| CD◠NN | 93.14 | P◠VV | 1.90 |

首先我们从一些统计数据上说明句法依存分析能潜在的帮助语义依存分析。我们对HIT语义依存树库和MALT句法依存树库的内部依存弧做一下统计比较，这一比较主要从两方面进行。第一个方面是考察弧的一致性对应情况，这个考察并不考虑弧上关系，只是考虑哪些依存弧在语义依存分析和句法依存分析中保持不变，进一步我们细化这个统计，考虑哪些词性对在这两种依存上面都是一致的；第二个方面是考察词的弧上关系，一般来说句法上面的主语往往会和语义上的主体语义角色(例如施事、经验者等等)比较相似，因此我们观察这样的弧上关系一致性是否存在。

对于弧的一致性，我们针对句法依存中的所有依存弧，去检查它们在语义依存中是否依然存在，然后细化，根据词性对去计算它们的一致性百分比。根据我们的语料统计显示，从整体上语义依存和句法依存的一致性比例达到60.13%。进一步，表3-4中显示了依存弧一致性和词性对的关系，也就是说哪些词性对更容易具有这种一致性或者更不容易具有这种一致性，我们列举了一致性最高的六个词性配对和一致性最低的六个词性配对，所有这些词性配对都在语料中出现了超过1,000次。对于一个词性配对，如果句法依存和语义依存的一致性非常高时，则当句法依存存在这个依存弧时，语义依存也更倾向于存在这个依存弧；反之如果弧一致性非常低时则语义依存更倾向于没有这个依存

弧。因此无论一致性特别高还是一致性特别低时，都可能会对语义依存分析有指导作用。

表 3-5 句法与语义弧上关系的对应关系。

Table 3-5 Label comparisons between semantic and syntactic dependencies.

| 句法标签 | 语义标签 |
|---|---|
| NMOD | d-attribute, d-restrictive, d-genetive, d-category, d-content |
| VMOD | aspect, patient |
| AMOD | d-deno, d-quantity, sequence, d-restrictive, degree |
| OBJ | content, isa, patient, |
| VC | state |
| DEP | d-sequence, d-deno |

同样，我们也分析了语义依存分析和句法依存分析弧上关系的对应规律，其最终结果如表3-5所示，我们列举了句法和语义标签对应比例比较高的部分例子。如果某个语义关系总是对应到某个句法关系时，那么这个句法关系就会为预测某个语义关系提供潜在的证据，从而能起到积极的帮助作用。

### 3.6.2 语义依存模型融入句法特征

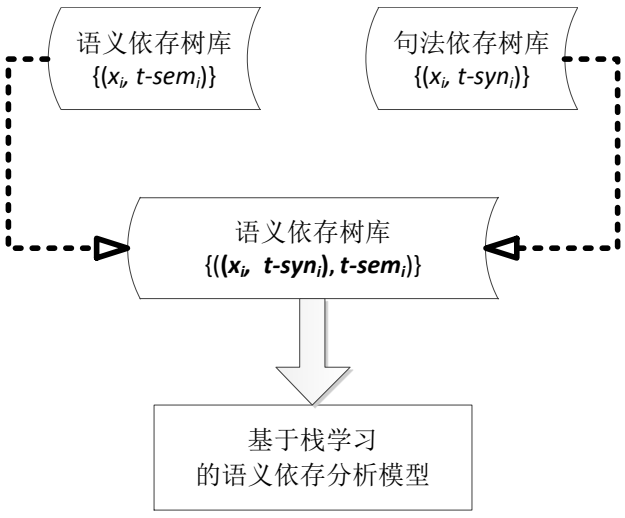



图 3-6 融入了句法依存特征的基于栈学习的语义依存模型框架图。

Fig. 3-6 The framework of a stacked semantic dependency parsing model with syntactic dependencies.

为了验证句法依存信息对语义依存信息的帮助作用，我们采用了基于栈学习(Stacking Learning)的方法将句法依存分析的结果融入到语义依存分析模型中[107]。栈学习的总体框架图如图3-6所示，首先我们利用一个依存句法分析器来对指定句子进行解码，得到其依存句法树，然后从这棵依存句法树中抽取特定的特征，加入到语义依存分析的模型中。该方法主要借鉴了李正华等人2012年提出的关于不同依存树库融合的方法[108]，我们简单的将语义依存树库和句法依存树库看作两种不同类型的树库，但是我们的模型与他们的方法存在着两个不同的地方。首先我们的语义依存树库和句法依存树库都建立在同样标注标准的相同句子上面，并不像他们的任务中两个树库的原始句子都完全不一样；另外我们采用的依存分析方法也不一样，因为语义依存标签数目众多，基于图的依存分析算法时间复杂度过高，速度很慢，因此我们使用了基于转移的依存分析算法。

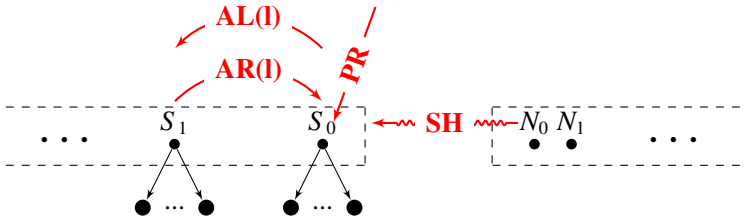



图 3-7 标准弧转移算法。

Fig. 3-7 The arc-standard transition algorithm.

我们采用基于转移的标准弧转移算法结合柱搜索来分别进行句法和语义依存分析。在该算法中，其转移系统由系统的状态和这个状态能接受的一系列操作组成，系统的状态由一个栈和一个队列组成，如图3-7所示，栈中存储着部分解码的依存树序列，而队列中存储着尚未处理的词。开始解码时，栈为空而队列中存储着一个句子所有的词，经过一系列转移操作后，系统进入终结状态，此时栈中仅含有一棵依存树而队列为空，栈中的依存树即为最终的解码结果。在标准弧转移系统中，一共定义了四种不同类型的转移动作，分别为

(1) SHIFT，简称SH：将队列中的第一个元素移入栈中，形成一个仅包含一个节点的部分解码依存树；

(2) ARC-LEFT(label)，简称AL(l)：将栈顶的两棵部分解码依存树进行合并，形成一个左弧，使得栈顶的第一棵依存树的根节点为合并后的新的部分解码依存树的根节点；

(3) ARC-RIGHT(label)，简称AR(l)：将栈顶的两棵部分解码依存树进行合

并，形成一个右弧，使得栈顶的第二棵依存树的根节点为合并后的新的部分解码依存树的根节点；

(4) POP-ROOT，简称PR：如果队列中的元素为空，而且栈中只包含一棵部分解码依存树时，将这个依存树移除，并令这棵依存树的根节点为全句分析的根节点。

任何依存句法树$d$，都可以唯一的由一组长度为$2n$的转移动作序列$A_1 \cdots A_{2n}$从初始状态转移到达最终状态，最终这棵句法树的分数可以由公式3-1计算得到。在公式中，$w$表示模型，$f$表示特征，$S$和$N$表示栈或者队列中的节点。

$$\text{Score}(\mathbf{d}) = \sum_{i=1}^{2n} w \cdot f(A_i, S, N) \tag{3-1}$$

对于句法依存分析模型，我们采用前人已经定义好的一些基本特征，如表3-6中的基本特征所示，其中$S$或者$N$的下标表示距离栈顶或者队列头部的相对位置，$w$和$t$表示某个节点词和词性，$s_l$和$s_r$ 表示某个节点所有左孩子或者右孩子的标签; 下标$l$和$r$代表某节点最左部或者最右部的孩子，下标$l2$和$r2$代表某节点最左边第二个或者最右边第二个孩子。而对于语义依存分析模型，我们除了抽取基本特征之外，还从自动分析的句法依存树中抽取了如表3-6中所示的指导特征(Guided features)，其中$h_{guide}$ 表示栈顶两个元素在句法依存树中所对应的弧一致性关系，主要分为三种：左弧一致性（栈顶的两个节点在句法树中是左弧关系）、右弧一致性（栈顶的两个节点在句法树中是右弧关系）和其它（栈顶的两个节点在句法树中没有依存弧存在）；$l_{guide}$ 表示同样的一个节点对应在句法依存树中的标签。在句法和语义依存分析的模型中，特征的权重通过平均感知器结合提前更新算法训练得到。

## 3.7 中文语义依存与句法依存联合模型

### 3.7.1 动机

前面的一节中我们主要介绍了中文句法依存分析对语义依存分析的潜在帮助作用，但是所使用的方法是基于特征融合的串行模型。由于串行模型会引入错误蔓延问题，也就是输入的句法依存树的错误会引入到语义依存分析中去，另一方面在级联模型中，句法依存分析无法利用语义依存分析的结果，因此句法依存分析的性能也无法达到最佳的状态从而不能更好的为语义依存分析服务，为了解决这两个问题，我们提出了一个语义依存分析和句法依存分析的联

表 3-6 语义依存和句法依存模型中使用的特征模板

Table 3-6 Feature templates of the semantic and syntactic dependency parsing models.

| **基本特征** |
| --- |
| $S_0w$ $S_0t$ $S_0wt$ $S_1w$ $S_1t$ $S_1wt$ $N_0w$ $N_0t$ $N_0wt$ $N_1w$ $N_1t$ $N_1wt$ |
| $S_0w{\cdot}S_1w$ $S_0w{\cdot}S_1t$ $S_0t{\cdot}S_1w$ $S_0t{\cdot}S_1t$ $S_0w{\cdot}N_0w$ $S_0w{\cdot}N_0t$ $S_0t{\cdot}N_0w$ $S_0t{\cdot}N_0t$ |
| $S_{0l}w$ $S_{0r}w$ $S_{0l}t$ $S_{0r}t$ $S_{0l}l$ $S_{0r}l$ $S_{1l}w$ $S_{1r}w$ $S_{1l}t$ $S_{1r}t$ $S_{1l}l$ $S_{1r}l$ |
| $S_{0l2}w$ $S_{0r2}w$ $S_{0l2}t$ $S_{0r2}t$ $S_{0l2}l2$ $S_{0r2}l2$ $S_{1l2}w$ $S_{1r2}w$ $S_{1l2}t$ $S_{1r2}t$ $S_{1l2}l2$ $S_{1r2}l2$ |
| $S_0t{\cdot}S_{0l}t{\cdot}S_{0l2}t$ $S_0t{\cdot}S_{0r}t{\cdot}S_{0r2}t$ $S_1t{\cdot}S_{1l}t{\cdot}S_{1l2}t$ $S_1t{\cdot}S_{1r}t{\cdot}S_{1r2}t$ |
| $S_0t{\cdot}S_1t{\cdot}S_{0l}t$ $S_0t{\cdot}S_1t{\cdot}S_{0l2}t$ $S_0t{\cdot}S_1t{\cdot}S_{0r}t$ $S_0t{\cdot}S_1t{\cdot}S_{0r2}t$ |
| $S_0t{\cdot}S_1t{\cdot}S_{1l}t$ $S_0t{\cdot}S_1t{\cdot}S_{1l2}t$ $S_0t{\cdot}S_1t{\cdot}S_{1r}t$ $S_0t{\cdot}S_1t{\cdot}S_{1r2}t$ |
| $S_0tv_r$ $S_0tv_l$ $S_0wv_r$ $S_0wv_l$ $S_1wv_l$ $S_1tv_l$ |
| $S_0ws_r$ $S_0ts_r$ $S_0ws_l$ $S_0ts_l$ $S_1ws_l$ $S_1ts_l$ |
| **指导特征** |
| $S_0w{\cdot}h_{guide}$ $S_0t{\cdot}h_{guide}$ $S_0wt{\cdot}h_{guide}$ $S_1w{\cdot}h_{guide}$ $S_1t{\cdot}h_{guide}$ $h_{guide}$ |
| $S_0w{\cdot}S_0l_{guide}$ $S_0t{\cdot}S_0l_{guide}$ $S_0wt{\cdot}S_0l_{guide}$ $S_1w{\cdot}S_0l_{guide}$ $S_1t{\cdot}S_0l_{guide}$ $S_0l_{guide}$ |
| $S_0w{\cdot}S_1l_{guide}$ $S_0t{\cdot}S_1l_{guide}$ $S_0wt{\cdot}S_1l_{guide}$ $S_1w{\cdot}S_1l_{guide}$ $S_1t{\cdot}S_1l_{guide}$ $S_1l_{guide}$ |

合模型去同时处理这两个任务，使得它们能够更好的进行交互。

语义依存分析和句法依存分析之所以有效，还有一个潜在动机在于中文句法和语义依存有很多一致的依存弧，前面一节中我们提到接近60.13%的依存弧在不考虑弧上关系时是完全一样的，这些相同的依存弧可以作为两者分析的一个桥梁，不一样的语义或者句法依存可以作为一个额外的附加证据来支持这些相同的依存弧，而联合模型中可以非常方便的利用这些证据。

### 3.7.2 方法

我们通过直接扩展基于转移的标准弧转移依存分析方法，来得到最终的联合模型。在基本的标准弧转移算法中，自动机转移系统是其核心模块，主要由状态和在状态基础上的一组转移动作所控制，其状态由一个栈和一个队列构成，栈中存储着部分解码的依存树序列，而队列中存储着尚未处理的词，转移系统中的动作共有四类，分别移进、左弧、右弧和移除根，它们的各自定义已经在前面一节中介绍。对于扩展后的联合模型，它也是一个自动机转移系统，其状态是对基本模型状态的一个简单扩充，由两个栈和两个队列组成，其中一

个栈存储着部分解码的语义依存树序列，另一个栈中存储着部分解码的句法依存树序列；其中的一个队列存储着尚未进行语义依存分析的词序列；而另一个队列存储着尚未进行句法依存分析的词序列。联合模型转移系统的动作也扩充为原来的两倍，其中一部分用来处理语义依存($\mathrm{SH}^{sem}$，$\mathrm{AL}^{sem}(l)$，$\mathrm{AR}^{sem}(l)$，$\mathrm{PR}^{sem}$)，另一部分用来处理句法依存($\mathrm{SH}^{syn}$，$\mathrm{AL}^{syn}(l)$，$\mathrm{AR}^{syn}(l)$，$\mathrm{PR}^{syn}$)。这些动作的定义和基准模型完全一样，除了每一类动作都必须应用在语义依存或者句法依存所对应的栈和队列上。图3-8简单的说明了我们的扩充方法，给出了扩展之后的状态和操作示意图。

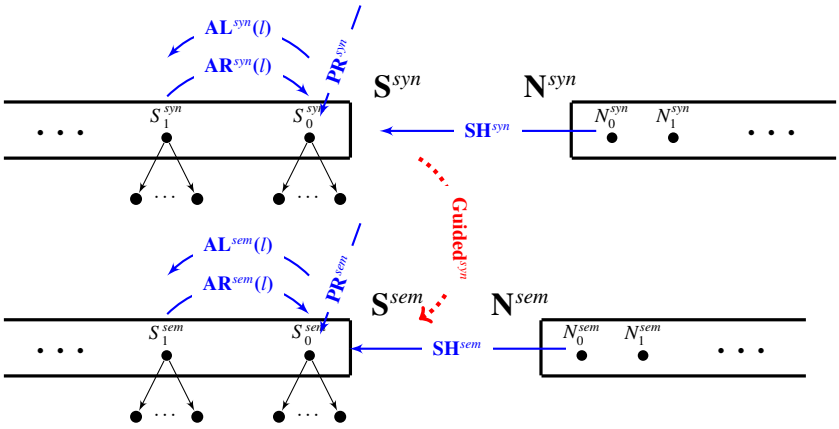



图 3-8 语义和句法依存联合模型状态以及动作转移示意图。

Fig. 3-8 Illustrations of the states and actions for the joint model.

在联合模型的转移系统中，虽然状态和动作都扩充为原来的两倍，但是却产生了一个新的问题。针对一个句子，我们假定使用原来的串行模型产生这个句子的句法依存树和语义依存树的动作转移序列分别为$A_1^{syn}A_2^{syn}\cdots A_n^{syn}$和$A_1^{sem}A_2^{sem}\cdots A_n^{sem}$，$\mathrm{ST}_0^{syn}\mathrm{ST}_1^{syn}\cdots\mathrm{ST}_n^{syn}$和$\mathrm{ST}_0^{sem}\mathrm{ST}_1^{sem}\cdots\mathrm{ST}_n^{sem}$分别为转移系统产生这两棵最终的依存树时中间经历的状态序列，那么在联合模型中最终的正确转移动作序列应该恰好包含$A_1^{syn}A_2^{syn}\cdots A_n^{syn}$和$A_1^{sem}A_2^{sem}\cdots A_n^{sem}$这两个序列。但是如果某一时刻联合模型的状态到达了$(\mathrm{ST}_i^{syn},\mathrm{ST}_j^{sem})$，那么下一个转移动作我们应该如何选取呢，是$A_{i+1}^{syn}$还是$A_{j+1}^{sem}$？在这一语义依存分析和句法依存分析的联合模型中我们规定了必须先解析句法依存树，待句法依存树分析完毕然后再解析语义依存树，也就是联合模型中正确的动作转移序列应该为$A_1^{syn}A_2^{syn}\cdots A_n^{syn}A_1^{sem}A_2^{sem}\cdots A_n^{sem}$。

在这个联合模型中，所使用的特征基本上包括三部分，第一部分和第二部分分别是根据表3-6中的基本特征抽取的语义依存分析和句法依存分析的特征，

第三部分是表3-6中的指导特征，在语义依存分析时根据句法依存树的信息抽取。同样，联合模型的特征权重训练方法也采用平均感知器结合提前更新算法。这一联合模型和第3.6节提到的串行模型很类似，只不过是使用了一个统一的模型把原有的两个串行模型结合在了一起，使得句法依存能更有效的指导语义依存，同时实际上这个模型也能调整句法依存模型的结果，一定程度上也能提高句法依存的性能。

## 3.8 实验

本章内容涉及到的实验包括四个部分，第一部分是比较HIT语义依存和MALT句法依存这两种自动依存分析器的性能，这两个自动分析器是使用同样句子集合的训练和开发语料，并且采用完全一样的依存分析模型(第3.6节中使用的基于转移的标准弧转移模型)而得到的；这一对比的目的在于表明语义依存分析的难度。第二部分实验是同样利用上面两种依存语料，建立相应的基于依存的语义角色标注模型，观察这两个模型在语义角色标注上的性能对比，以验证语义依存对于语义角色标注的有效性，从而以实验的方式表明语义依存分析的合理性。第三部分实验是为了验证MALT这种面向句法的依存分析是否能够帮助语义依存分析，从而为最终提出的联合模型做铺垫。最后一部分的实验为了验证我们最后提出的语义依存分析和句法依存分析的联合模型的有效性。

对于所有上面的实验，HIT语义依存和MALT句法依存语料的原始句子采用的是SemEval-2012语义依存评测中的句子，其训练、开发以及测试语料的划分方法也是按照评测中的方式来进行的[57]，相关语料统计信息如表3-7所示。HIT语义依存分析的语料就是SemEval-2012语义依存评测中的语料，而MALT句法依存语料是根据这些句子找到它们在中文宾州树库中的短语结构句法树然后通过相应的规则转换而得到的。

依存分析的评价指标采用了标准的基于词的不带标签依存弧准确

表 3-7 实验中所用到的语料统计信息。
Table 3-7 Corpus statistics in our experiments.

| | 句子数 | 词数 | 谓词数 |
|---|---|---|---|
| Train | 8,301 | 250,311 | 42,124 |
| Devel | 534 | 15,329 | 2,353 |
| Test | 1,233 | 34,311 | 5,246 |

率(Unlabeled Attachment Score, UAS) 、带标签的依存弧准确率(Labeled Attachment Score, LAS)以及带标签的句子准确率(Completely Match, CM)这三个方法。第二部分的实验中，语义角色标注语料是从Chinese Propbank 2.0(CPB2)中提取出来的[51]，因为名词语义角色数量比较少，所以我们只使用了动词谓词的那部分语料。语义角色标注的性能评价采用语义角色块的识别准确率(Precision，P)、召回率(Recall, R)以及它们的F-measure值(F)这三个指标。

### 3.8.1 自动依存分析性能对比

首先我们来观察语义依存和句法依存的自动分析难度对比，表3-9中的基准模型部分显示了自动语义依存分析和自动句法依存分析的结果。在基于同样的训练、开发和测试句子集合上，最终HIT语义依存分析的性能要远远低于MALT句法依存分析的性能，其中UAS相差接近4%，而LAS相差接近20%，这一结果和我们第3.4节中的分析一致，即自动语义依存分析的难度要远高于自动句法依存分析。

### 3.8.2 语义角色标注性能对比

表 3-8 语义角色标注的最终结果。

Table 3-8 Final SRL results.

| 系统 | P | R | F |
|---|---|---|---|
| HIT | **82.65** | **75.36** | **78.84** |
| MALT | 80.89 | 75.12 | 77.90 |

这里我们观察在基于依存的语义角色标注系统中，是使用HIT语义依存还是使用MALT句法依存能带来更好的语义角色标注性能? 这两种基于依存分析的语义角色标注系统我们分别用HIT和MALT代表以便进行简单的区分。表3-8显示了这两个个系统的语义角色标注性能。从表中我们可以看到，HIT语义依存树能带来最好的语义角色标注性能，这也实际上表明了语义依存分析和语义角色标注之间密切的关系，说明了语义依存分析作为一个语义分析手段的合理性。

为了更清楚的理解这种性能提升，我们进一步展开了错误分析实验，来观察不同语义角色的性能在这两个系统中的变化。我们选取了出现频率比较高的七种语义角色来进行分析，包括三种核心语义角色A0，A1，A2和四种功能语义角色，状语修饰(Adverbial，ADV)，篇章连接词(Discourse connective，DIS)，

地点(Location，LOC)和时间(Temporal，TMP)。图3-9中显示了这个性能对比。从图中可以看到，对于大部分语义角色标注标签，语义依存都能带来更好的性能，但是对于A0标签，却有一定程度的下降，表明了语义依存分析和语义角色标注在A0上的不一致。这个不一致也是可以理解的，因为对于一些经验性动词的主体与客体的区别，是很难定义清楚的，例如“这里**存在**着一群好事的人”，对于“存在”的主体是比较难界定的。我们通过观察实验数据也发现了这些细微的区别。

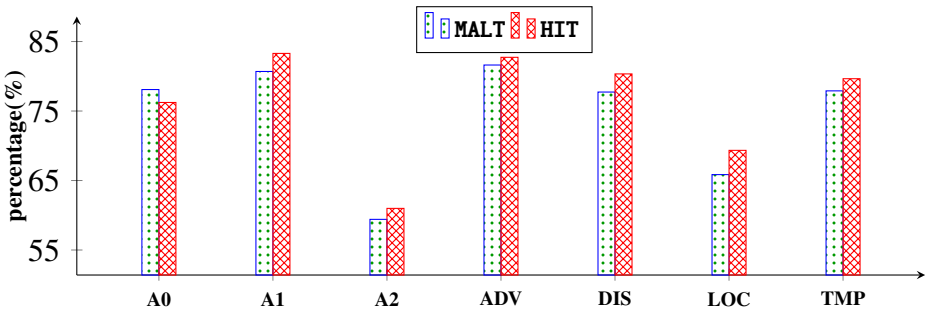



图 3-9 两种语义角色标注系统在不同语义角色上的性能。

Fig. 3-9 Performances of the two SRL systems with respect to semantic roles.

### 3.8.3 语义依存模型中融入句法特征

在验证了语义依存分析的合理性之后，我们来探讨句法依存信息能否为语义依存分析带来更好的性能，这样为我们最终提出的语义依存分析和句法依存分析的联合模型做出铺垫。在训练基准的HIT语义依存模型时，我们直接使用对应的训练和开发语料，而在训练基于栈学习的融入了MALT句法特征的HIT语义依存分析模型时，我们采用5交叉验证的方式来构造含有自动MALT句法树的语义依存训练语料，这个交叉验证模型就是用基准标准弧转移算法训练得到的，这一点是因为我们在测试时使用的MALT句法树是自动的，采用这种方法构造训练语料能使得训练环境和测试环境更相似，从而能获得更好的性能。

最终的实验结果如表3-9所示，其中ICT代表熊皓和刘群在参加SemEval2012评测时系统的结果[94]，Zhijun Wu表示Zhijun wu等人在SemEval2012评测时系统的结果[92]，Zhou qiaoli表示Zhou qiaoli等人在SemEval2012评测时系统的结果[91]，NJU代表Tang Guangchao等人参加SemEval2012评测时系统的结果[93]。从表中的结果可以看出，我们的基准模型已经取得了非常好的性能，而且在融入了句法特征后，能得到进一

步0.40%的性能提升。

### 3.8.4 语义和句法依存联合模型

表 3-9 句法和语义依存在测试数据集上的最终结果。

Table 3-9 Final syntactic and semantic dependency parsing results on the test data set.

| Model | MALT句法依存分析 | | | HIT语义依存分析 | | |
|---|---|---|---|---|---|---|
| | UAS | LAS | CM | UAS | LAS | CM |
| 基准模型 | 84.12 | 82.25 | **30.25** | 80.53 | 62.72 | 24.57 |
| 基于栈学习的串行模型 | 84.08 | 82.20 | 29.58 | 80.99 | 63.12 | 25.03 |
| 联合模型 | **84.22** | **82.38** | 29.76 | **81.59** | **63.55** | **26.12** |
| ICT | — | — | — | 80.45 | 62.80 | — |
| Zhijun Wu | — | — | — | 78.69 | 62.72 | — |
| Zhou qiaoli | — | — | — | — | 62.08 | — |
| NJU | — | — | — | 80.29 | 61.64 | — |

最后一部分的实验是为了验证我们最终提出语义依存分析和句法依存分析的联合模型，表3-9中显示了最终联合模型的性能，并且和其它相关模型进行了对比。其中基准模型是语义依存和句法依存进行独自训练所得到的模型，两者互不相关；基于栈学习的级联模型是前面一节提到的模型，其中的句法依存分析模型是我们利用语义依存分析的结果结合栈学习来得到的。最终的实验结果表明了我们提出的语义依存分析和句法依存分析的联合模型取得了最好的性能，不仅只是超过了基准模型的性能，而且也超过了基于栈学习的串行模型的性能。

为了更好的理解句法依存分析对语义依存分析的促进作用，我们对比联合模型和基准模型的最终测试结果，来观察哪种类型的语义依存或者句法依存得到了更大的帮助。联想到最初联合模型的动机，我们了解到那些在语义依存和句法依存中保持一致的弧由于能获取更多的证据而最有可能得到更好的结果。我们通过实验数据分析来验证这一观点，具体结果如表3-10所示，其中粗体部分部分表示性能一致和不一致的依存，谁在联合模型中具有更好的性能提升。从表中我们可以看到，显然一致的弧提升是比较明显的，相反，不一致的弧在联合模型中基本上是呈现性能下降趋势，这一分析结果完全符合联合模型被提出来的初始动机。

表 3-10 一致和不一致弧的性能对比。

Table 3-10 Performances of the consistent and inconsistent dependencies.

| | MALT句法依存分析 | | | | HIT语义依存分析 | | | |
|---|---|---|---|---|---|---|---|---|
| | 一致的弧 | | 不一致的弧 | | 一致的弧 | | 不一致的弧 | |
| | UAS | LAS | UAS | LAS | UAS | LAS | UAS | LAS |
| 基准模型 | 87.52 | 85.44 | 79.00 | 77.44 | 85.59 | 66.63 | 72.89 | 56.81 |
| 联合模型 | 87.81 | 85.74 | 78.82 | 77.32 | 87.36 | 68.08 | 72.87 | 56.72 |
| 差值 | **+0.29** | **+0.30** | -0.08 | -0.12 | **+1.77** | **+1.45** | -0.02 | -0.09 |

## 3.9 本章小结

在本章中，我们首先描述了语义依存分析的概念，然后和句法依存分析做了简单的对比，包括任务目的、标注规范以及自动分析难度。虽然语义依存分析能表达更多的信息，但是自动语义依存分析的难度也比句法依存分析高出很多，我们从实验上面也验证了这一点，其背后的原因在于语义依存的弧上关系数目以及语义依存的长距离弧数目都比句法依存多。

紧接着我们对语义依存分析作为一种语义分析手段的合理性进行了调查研究，首先我们对语义依存分析和语义角色标注的表示体系做了对比，说明了语义依存分析能表示比语义角色标注更多的信息。进一步我们还比较了句法依存和语义依存在语义角色标注上的性能对比，由于语义依存能带来更好的语义角色标注性能，从而说明了语义依存分析和语义角色标注更相关，而语义角色标注是目前被广泛认可的一种语义分析方法，因此这一实验也间接表明了语义依存分析的合理性。

然后我们讨论了语义依存和句法依存的一些统计性规律，借此表明句法依存分析可以潜在的帮助语义依存分析，我们提出了一种基于栈学习的融入了句法依存信息的语义依存模型来验证这一观点。最后我们提出了语义依存分析和句法依存分析的联合模型，最终的结果表明，联合模型能取得最好的语义依存分析性能，实验分析结果表明了联合模型能有效的帮助语义依存和句法依存中一致性比较高的那部分弧。

# 第 4 章 高效率高性能的词性句法联合模型

## 4.1 引言

在依存句法分析模型中，词性相关的特征是其中最有效的特征之一。在实际的自然语言处理中，词性标注和依存句法分析往往作为两个单独的任务进行处理，即首先进行词性标注，然后在此基础之上，进行依存句法分析，这种方法称为串行的分析方法。最近，部分研究者开始尝试将词性标注和依存句法联合起来，用一个统一的模型将词性标注和依存句法的结果同时分析出来。这种联合处理的方法取得了一定的成功。

词性标注是一个典型的序列标注问题，依存句法分析的主要方法有两种：基于图的依存句法分析和基于转移的句法分析。因此词性标注和依存句法分析的联合模型要么通过基于图的依存句法分析进行扩展，要么通过基于转移的依存句法分析进行扩展，前者被称为基于图的联合模型[4]，而后者被称为基于转移的联合模型[64]。这两类联合模型的方法都能取得很好的效果。

短语结构句法分析的方法，例如采用非词汇化的概率上下文无关文法进行处理的方法[30]，将词性标注看作短语结构句法分析的一部分，这样短语结构句法分析会同时也将词性标注的结果分析出来。另外，如果我们能将短语结构句法分析结果通过一些规则转换成依存句法，则可以借助于短语结构句法分析得到词性标注和依存句法分析的结果。这一类方法被称为基于短语结构的联合模型。

综上所述，对于词性标注和依存句法的联合模型，存在着三种不同的方法，这三种方法分别从各自的角度出发产生了最终的分析结果，各有利弊。因此我们可以进一步将这三种模型融合起来，得到一个更高性能的联合模型。另外，我们对融合模型以及三个基本模型进行分析，并与串行模型进行比较，这样分析出来的共性结果比任何一个单一的模型更能全面的反映中文词性标注与依存句法分析之间的关系，这体现了融合模型的第二个优点。

我们采用基于栈学习(Stacked Learning)的方法来融合这三个基本联合模型。相比其他模型融合方法，基于栈学习的融合模型形式优美简洁，而且其最终性能不局限于三个基本模型的结果之中，即便在错误的基本模型结果上，基于栈学习的融合模型也有可能得到正确的答案。在自然语言处理中，使用基于栈学

习的方法来进行模型融合已经有不少成功的例子，例如基于子词(sub-word)解码的分词词性标注联合模型，词法分析[60]，以及基于图和基于转移相结合的依存句法分析[99, 109, 110]。

另外值得关注的一点是，联合模型的解码效率问题使得它并不适合于应用到实际场景中。一般情况下，为了达到一个高水平的性能，基于图的联合模型在经过很多琐碎的裁剪优化之后解码能达到一句每秒，基于转移的联合模型能达到9句每秒，基于短语句法结构的能达到6句每秒。基于转移的联合模型采用的是一种基于柱搜索的解码算法，因此我们可以通过调整柱的大小非常方便对该模型做一个速度和性能上的平衡调整。根据前人的工作，为了达到一个理想性能，柱大小一般被设置为64，此时联合模型的解码速度接近9句每秒。如果我们将柱大小设置为2，则解码的速度可以提升接近32倍，也就是联合模型的解码速度能接近200句每秒，但是这一情况下性能却比柱大小为64的模型低了接近2.5%。如果我们能将这一柱大小为2的联合模型提升到和柱大小64的模型一个水平，便可以得到一个高效率高性能的词性标注依存句法联合模型。

本章我们将采用一种过训练(Up-training)的方法来提升柱大小为2的基于转移的简单联合模型的性能，使得其最终的性能能和普通柱大小为64的联合模型性能相当。其原理是利用一个复杂模型（基于栈学习的融合模型）为这一简单模型产生自动的大规模训练语料[111]，然后加入这一简单模型的训练语料中，使得简单模型的性能大大增强。

本章我们将逐步介绍这种速度快而且性能高的词性标注依存句法联合模型。我们首先介绍最相关的研究工作，然后描述三类基本的词性标注依存句法联合模型，紧接着提出基于栈学习的融合联合模型，这样我们便得到一个高准确率但是速度非常慢的复杂联合模型。进一步，基于这个复杂模型，我们使用过训练的方法来使得柱大小为2的基于转移的简单联合模型的性能提升一个级别，从而得到了我们最终的高效率高性能的联合模型。最后我们通过实验来逐步验证我们的方法。

## 4.2 相关工作

**1. 词性标注依存句法联合模型**

李正华等人最早提出了中文词性标注和依存句法联合模型[4]，他们在基于图的依存句法分析基础上对解码算法进行扩展，使得词性不再作为输入，而是作为输出，这样词性标注和依存句法分析的结果便同时被解析出来。进一步，

Jun Hatori等人在基于转移的依存句法分析上，通过对标准弧转移算法中的移进操作进行扩展，得到了一个基于转移的联合模型[64]。在这一模型中，每次移进一个中文词时，被移进词的词性成为了移进操作的参数，因此词性和依存句法的结果也同时被解析出来。

基于短语句法结构的依存句法分析很早就被人重视了[112]，而且其性能也不弱于基于图或者基于转移的依存句法分析。其基本原理是利用一些短语结构句法分析器将词性节点看作一种特殊的短语节点，从而词性标注的结果也被同时解析了出来，这样我们仅需应用短语结构转依存的规则将短语结构转换成依存结构，便同时得到了词性标注和依存句法分析的分析结果。

本章对这三种不同的方法进行比较，进一步融合，首先得到一个高性能但是低效率的复杂联合模型；然后利用该融合模型自动分析大量未标注语料，将其结果加入到一个效率高的简单联合模型的训练语料中，使得简单模型的性能得到增强，从而得到了一个性能和原来基本模型相当，但是速度却快了至少10倍以上的联合模型。

**2. 基于栈学习的模型融合方法**

基于栈学习的模型融合方法最早在1992年被David H. Wolpert提出[107]，该方法进一步被Leo Breiman在1996年加以论证[113]。它的核心思想是从多个模型中选取一个作为高层(第二层)最终模型，其它模型作为底层(第一层)模型，首先从底层获取若干结果，然后提供给高层模型作为输入，并转化成为特征使得高层模型的性能得到加强。

基于栈学习的模型融合方法在自然语言处理中有广泛的应用，这里仅仅列出最相关的和依存句法模型融合相关的文章。Ryan McDonald在他的博士毕业论文中最早使用了基于栈学习的模型融合方法[112]，他先后利用两种不同的短语句法结构分析器Collins parser [22] 和Charniak parser [114]，分析出短语句法结构，进一步将结果转换为依存结构，然后分别把它们的结果融入到一个二阶的基于图的依存句法分析器中，最终实验结果表明了该方法的有效性。在ACL2008年的论文中，Ryan McDonald和Joakim Nivre使用基于栈学习的模型融合方法将基于图的和基于转移的依存句法分析融合起来[109]，得到了很好的性能提升；进一步他们对融合方法做了仔细分析和扩充，相关的结果发表在2011年的计算语言学期刊上[99]。同样在2008年，A.F.T Martins等人也采用了基于栈学习的方法来融合基于图的和基于转移的依存句法分析[110]。

本章所面对的情况更为复杂，首先我们将基于栈学习的模型融合方法应用在一个联合模型上面，面临多个任务；其次我们的基本模型数目并不是简单的

两种不同的模型的融合，而是三种模型的融合，这样基于栈学习的模型融合时，在高层模型中存在两个输入结果，两者的相互比较信息也会对模型的融合有一定的帮助。

**3. 过训练方法**

过训练(Up-training)方法最早由Petrov Slav等人提出，用于提升问句的句法分析性能[111]。它的核心是使用某个特定的复杂模型来为一个简单模型自动生成一些训练语料，该复杂模型要么解码方法更为复杂，速度非常慢，要么它能利用到一些一般情况下无法使用的约束条件，这样该复杂模型的性能会比简单模型要好很多。我们将这个复杂模型自动解码生成的训练语料加入到一个简单模型的训练语料中，使得简单模型的性能大大增强。

车万翔等人2013年提出的使用双语数据来改善命名实体识别的性能便可以看作是一种过训练[115]，其复杂模型需要使用双语的特征，但是该复杂模型无法用于一般的情况，因为并非所有的现实句子都能为其找到对应的双语翻译。因此他们使用了过训练的方法来提升单语简单命名实体识别模型的性能。在本章的研究中，我们所使用的复杂模型为一个速度非常慢（解码速度不到1句每秒）的融合联合模型，而简单模型的解码速度能达到100句每秒以上，因此我们使用过训练的目的是提升联合模型的速度。

## 4.3 基准模型

词性标注和依存句法的联合模型，其目的在于对任意指定的句子$w_1 \cdots w_n$，为其自动分析出一组最优的词性序列$t = t_1 \cdots t_n$和一棵最优的依存句法树$d$。本节我们将介绍三种不同的基本联合模型方法，分别为基于图的联合模型算法(JGraph)，基于转移的联合模型算法(JTrans)和基于短语结构的联合模型算法(JConst)，这三种方法对词性标注和依存分析的建模方式各不一样，性能以及最终分析结果的错误分布也互不相同。

### 4.3.1 基于图的联合模型算法

基于图的联合算法，最先由李正华等人于2011年提出[4]，后续他们进一步还对这个工作进行了优化，相应的工作发表在COLING2012上面[61]，我们用JGraph来表示这种方法。对于给定的句子$\mathbf{x}$，基于图的联合模型根据公式4-1为该句子搜索到一棵句法依存树$\mathbf{d}$以及词性序列$\mathbf{t} = t_1 \cdots t_n$，使得其分数最高：

$$\mathrm{Score}_{\mathrm{joint}}(\mathbf{x}, \mathbf{t}, \mathbf{d}) = \mathbf{w}_{\mathrm{pos}} \cdot \mathbf{f}_{\mathrm{pos}}(\mathbf{x}, \mathbf{t}) + \mathbf{w}_{\mathrm{dep}} \cdot \mathbf{f}_{\mathrm{dep}}(\mathbf{x}, \mathbf{t}, \mathbf{d}) \quad (4\text{-}1)$$

其中**f**表示特征向量，**w**表示特征权重向量。我们一般使用平均感知器算法来学习特征的权重。

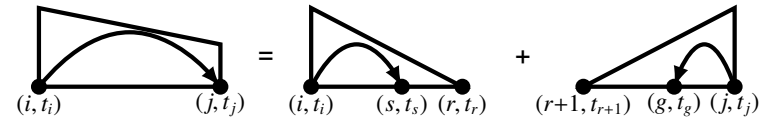
一般情况下，为了降低搜索算法的时间复杂度，首先要将一棵依存树**d** 进行分解成若干子树(Subtree)，然后将所有子树对应的分数进行累加，便得到了这棵依存树**d** 和相应词性**t** 的总分数。根据子树所包含边数目的不同，基于图的联合模型可以采用一阶、二阶或者高阶解码算法，其中一阶算法中最小子树仅包含一条依存弧，二阶算法中的最小子树包含两条依存弧，依次类似扩展到高阶解码算法。阶数的大小也决定了模型所能使用的特征，阶数越高，能使用的特征也会越多，但是解码速度也会更慢。

我们通过权衡速度和性能之后，采用Carreras 等人提出的二阶动态规划解码算法[32]。该算法定义了一种称为SPAN的基本数据结构，一个SPAN可以看做覆盖一组词语的部分解码依存句法树。任何一个SPAN可以是一个完整的SPAN(如图4-1 中(c)和(d) 的左边部分) 或者是一个不完整的SPAN(如图4-1 中(a) 和(b) 的左边部分)。对于完整的SPAN，其一侧的所有词(父亲节点除外)的所有孩子以及父亲节点都已经找到，而对于不完整的SPAN，头尾词以外的所有词的孩子节点和父亲节点都已经找到。二阶解码算法的实际过程就是SPAN之间相互组合逐渐变大的过程，具体SPAN之间的相互合并过程如图4-1 所示。在该图中，一共描述了四种不同类型SPAN的生成算法，包括两种不完整SPAN($I[(i,t_i),(j,t_j),i\curvearrowright j]$，$I[(i,t_i),(j,t_j),i\curvearrowleft j]$) 和两种完整SPAN($C[(i,t_i),(j,t_j),(r,t_r),i\curvearrowright r]$，$C[(i,t_i),(j,t_j),(r,t_r),i\curvearrowleft r]$])，其中一个不完整的SPAN由两个完整的SPAN加上一个依存弧生成，而一个完整的SPAN由一个不完整的SPAN和一个完整的SPAN生成。

对于句子$w_1\cdots w_n$，解码最初始时，对于任意的$i \in [1,n]$定义$C[(i,t_i),(i,t_i),(i,t_i),i\curvearrowright i] = 0$ 以及$C[(i,t_i),(i,t_i),(i,t_i),i\curvearrowleft i] = 0$，然后我们利用图4-1中四种合成方法分别计算$I[(i,t_i),(i+1,t_{i+1}),i\curvearrowright(i+1)]$，$I[(i,t_i),(i+1,t_{i+1}),i\curvearrowleft(i+1)]$，$C[(i,t_i),(i+1,t_{i+1}),(i+1,t_{i+1}),i\curvearrowright(i+1)]$ 和$C[(i,t_i),(i+1,t_{i+1}),(i,t_i),i\curvearrowleft(i+1)]$，上面的计算相当于SPAN覆盖的词数目由1变为了2，进一步同样利用这四种合成方法逐步计算覆盖词数目更大的SPAN，$I[(i,t_i),(j,t_j),i\curvearrowright j]$，$I[(i,t_i),(j,t_j),i\curvearrowleft j]$，$C[(i,t_i),(j,t_j),(r,t_r),i\curvearrowright r](i < r \le j)$ 和$C[(i,t_i),(j,t_j),(r,t_r),j\curvearrowleft r](i \le r < j)$，直到SPAN覆盖所有的词，最后的解码结果就是使得$C[(0,\#),(n,t_n),(r,t_r),0\curvearrowleft r]$最大的那个$r$对应的SPAN，从这个SPAN中可以直接提取出依存句法树**d**和相应的词性序列**t**。

在上述的二阶解码算法中，所使用的具体特征如表4-1 所示，其中$w$表示

**(a)**




$$I[(i,t_i),(j,t_j),i\curvearrowright j] = \max_{i\le r<j} \max_{(t_r,t_{r+1})\in\mathcal{T}^2} \{$$

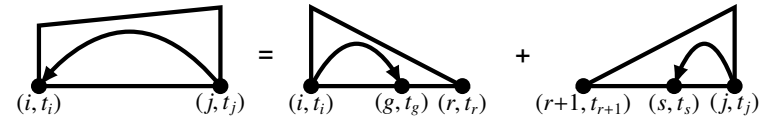
$$\max_{i\le s\le r, t_s\in\mathcal{T}} \{C[(i,t_i),(r,t_r),(s,t_s),i\curvearrowright s] + \text{Score}_{\text{sib}}(\mathbf{x}, t_i, t_j, t_s, i\curvearrowright j, i\curvearrowright s)\}$$

$$+ \max_{r+1\le g<\le j, t_g\in\mathcal{T}} \{C[(r{+}1,t_{r+1}),(j,t_j),(g,t_g),j\curvearrowright g]$$

$$+ \text{Score}_{\text{grd}}(\mathbf{x}, t_i, t_j, t_g, i\curvearrowright j\curvearrowright g)\}$$

$$+ \text{Score}_{\text{dep}}(\mathbf{x}, t_i, t_j, i\curvearrowright j) + \text{Score}_{\text{posuni}}(\mathbf{x}, j, t_j) + \text{Score}_{\text{posbi}}(\mathbf{x}, r+1, t_r, t_{r+1})\}$$

**(b)**




$$I[(i,t_i),(j,t_j),j\curvearrowright i] = \max_{i\le r<j} \max_{(t_r,t_{r+1})\in\mathcal{T}^2} \{$$

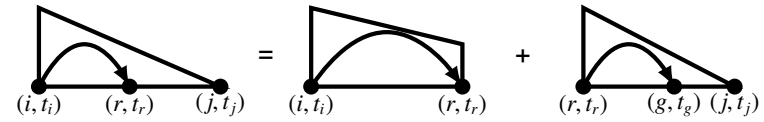
$$\max_{i\le s\le r, t_s\in\mathcal{T}} \{C[(i,t_i),(r,t_r),(g,t_g),i\curvearrowright g] + \text{Score}_{\text{grd}}(\mathbf{x}, t_j, t_i, t_g, j\curvearrowright i\curvearrowright g)\}$$

$$+ \max_{r+1\le g<\le j, t_s\in\mathcal{T}} \{C[(r{+}1,t_{r+1}),(j,t_j),(s,t_s),j\curvearrowright s]$$

$$+ \text{Score}_{\text{sib}}(\mathbf{x}, t_j, t_s, t_i, j\curvearrowright i, j\curvearrowright s)\}$$

$$+ \text{Score}_{\text{dep}}(\mathbf{x}, t_j, t_i, j\curvearrowright i) + \text{Score}_{\text{posuni}}(\mathbf{x}, j, t_j) + \text{Score}_{\text{posbi}}(\mathbf{x}, r+1, t_r, t_{r+1})\}$$

**(c)**

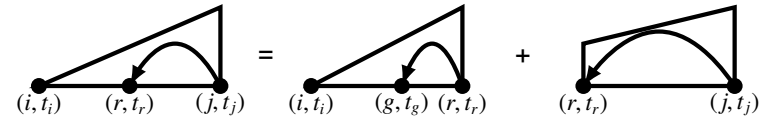



$$C[(i,t_i),(j,t_j),(r,t_r),i\curvearrowright r] = \max_{r\le g\le j} \max_{t_g\in\mathcal{T}} \{$$

$$I[(i,t_i),(r,t_r),i\curvearrowright r] + C[(r,t_r),(j,t_j),(g,t_g),r\curvearrowright g]$$

$$+ \text{Score}_{\text{grd}}(\mathbf{x}, t_i, t_r, t_g, i\curvearrowright r\curvearrowright g)\}$$

**(d)**




$$C[(i,t_i),(j,t_j),(r,t_r),j\curvearrowright r] = \max_{r\le g\le j} \max_{t_g\in\mathcal{T}} \{$$

$$I[(r,t_r),(j,t_j),j\curvearrowright r] + C[(i,t_i),(r,t_r),(g,t_g),r\curvearrowright g]$$

$$+ \text{Score}_{\text{grd}}(\mathbf{x}, t_j, t_r, t_g, j\curvearrowright r\curvearrowright g)\}$$

图 4-1 基于图的联合模型SPAN操作示意图。

Fig. 4-1 Span combinations for the graph-based joint model.

表 4-1 基于图的联合模型的特征模板。

Table 4-1 Feature templates for the graph-based joint model.

| 类别 | 特征模板 |
|---|---|
| 词性标注 | $t_i w_i$ $t_i w_{i-1}$ $t_i w_{i-2}$ $t_i w_{i+1}$ $t_i w_{i+2}$ $t_i w_{i-1} w_i$ $t_i w_i w_{i+1}$ $t_i w_{i-1} w_{i+1}$<br>$t_i t_{i+1}$ $t_i \cdot prefix(w_i, k)$ $t_i \cdot suffix(w_i, k)$ 其中$1 \le k \le 3$ |
| 依存句法分析 | $w_h \cdot dir \cdot dist$ $t_h \cdot dir \cdot dist$ $w_m \cdot dir \cdot dist$ $t_m \cdot dir \cdot dist$<br>$w_h t_h \cdot dir \cdot dist$ $w_m t_m \cdot dir \cdot dist$<br>$w_h t_h w_m dir \cdot dist$ $t_h w_m t_m \cdot dir \cdot dist$<br>$w_h w_m t_m \cdot dir \cdot dist$ $w_h t_h t_m \cdot dir \cdot dist$<br>$w_h t_h w_m t_m \cdot dir \cdot dist$ $t_h t_m \cdot \#\mathrm{punc}(h, m) \cdot dir \cdot dist$<br>$t_h t_m \cdot dir \cdot dist$ $w_h t_m \cdot dir \cdot dist$ $w_h t_m \cdot dir \cdot dist$ $w_h w_m \cdot dir \cdot dist$<br>$t_h t_{h+1} t_{m-1} t_m \cdot dir \cdot dist$ $t_h t_{h+1} t_m t_{m+1} \cdot dir \cdot dist$ $t_{h-1} t_h t_{m-1} t_m \cdot dir \cdot dist$<br>$t_{h-1} t_{h+1} t_{m-1} t_m \cdot dir \cdot dist$ $t_{h-1} t_h t_{h+1} t_m \cdot dir \cdot dist$ $t_h t_{m-1} t_m t_{m+1} \cdot dir \cdot dist$<br>$t_h t_{h+1} t_m \cdot dir \cdot dist$ $t_h t_{m-1} t_m \cdot dir \cdot dist$ $t_h t_m t_{m+1} \cdot dir \cdot dist$<br>$t_h t_s t_m \cdot dir \cdot dist$ $w_h t_s t_m \cdot dir \cdot dist$ $t_h w_s t_m \cdot dir \cdot dist$ $t_h t_s w_m \cdot dir \cdot dist$<br>$t_s t_m \cdot dir \cdot dist$ $w_s w_m \cdot dir \cdot dist$ $t_s w_m \cdot dir \cdot dist$ $w_s t_m \cdot dir \cdot dist$<br>$t_s t_{s+1} t_m \cdot dir$ $t_{s-1} t_s t_m \cdot dir$ $t_s t_{m-1} t_m \cdot dir$ $t_s t_m t_{m+1} \cdot dir$<br>$t_s t_{s+1} t_{m-1} t_m \cdot dir$ $t_{s-1} t_s t_{m-1} t_m \cdot dir$ $t_s t_{s+1} t_m t_{m+1} \cdot dir$ $t_{s-1} t_s t_m t_{m+1} \cdot dir$<br>$t_g t_m \cdot dir \cdot gdir$ $t_g t_h t_m \cdot dir \cdot gdir$ $t_g t_m t_{m+1} \cdot dir \cdot gdir$<br>$w_g t_m \cdot dir \cdot gdir$ $w_g w_m \cdot dir \cdot gdir$ $t_g w_m \cdot dir \cdot gdir$ $t_g w_h t_m \cdot dir \cdot gdir$<br>$w_g t_h t_m \cdot dir \cdot gdir$ $t_g t_h w_m \cdot dir \cdot gdir$ $t_g t_{m-1} t_m \cdot dir \cdot gdir$<br>$t_{g-1} t_g t_{m-1} t_m \cdot dir \cdot gdir$ $t_{g-1} t_g t_m \cdot dir \cdot gdir$ $t_g t_{g+1} t_m \cdot dir \cdot gdir$<br>$t_{g-1} t_g t_m t_{m+1} \cdot dir \cdot gdir$ $t_g t_{g+1} t_{m-1} t_m \cdot dir \cdot gdir$ $t_g t_{g+1} t_m t_{m+1} \cdot dir \cdot gdir$ |

词，$t$表示词性，$prefix(w,k)$和$suffix(w,k)$分别表示词$w$长度为$k$的前缀和后缀，$h$表示其父亲节点对应的序号，$m$表示孩子节点对应的序号，$dir$表示表示依存弧的方向，$gdir$表示某依存弧的父亲和其父亲的父亲所处的依存弧的方向，$dist$表示某个依存弧的父亲节点和该节点的距离，#punc$(h,m)$表示词$h$和词$m$之间所包含的标点符号的数目。

### 4.3.2 基于转移的联合模型

基于转移的联合模型最早由Jun Hatori等人于2011年提出并发表在IJCNLP上[64]，我们用JTrans来表示这一联合模型。该方法借鉴了自动机的思想，其核心模块由一个转移系统组成，转移系统由系统的状态和这个状态能接受的一系列操作组成，在开始解码时，有一个初始的状态，经过一系列转移操作后，系统进入终结状态，任何一个终结状态对应为一棵依存句法树，这棵依存句法树可以由中间的经历的转移操作序列直接得到。

对于我们的词性标注依存句法联合模型，系统的状态由一个栈和一个队列组成，栈中是部分解码的依存句法子树序列，记为$S_0, S_1, \cdots$，队列中是需要进一步处理的词语序列，记为$Q_0, Q_1, \cdots$。初始状态时，栈为空，队列中为$w_1, w_2, \cdots, w_n$，而终结状态时，栈中仅有一棵依存句法树，队列为空。在系统的状态上定义的操作有两类，移进和归约，两类动作均带有参数。对于移进，参数是词性，将队列中的第一个词赋予词性并移入栈中；而对于归约，实际上就是将栈顶的两棵子依存树进行合并，其参数主要是为了说明该归约是左归约还是右归约，左归约后栈顶的第二棵树将成为第一棵树的孩子节点，而右归约后栈顶的第一棵树将成为第二棵树的孩子节点。如图4-2所示显示了联合模型情况下的转移系统，最上面的是转移系统的状态，下面分别表示经过移进和归约之后状态的变化情况。

任何依存句法树$\mathbf{d}$及其相应的词性标记$\mathbf{t}$，都可以唯一的由一组长度为$2n$的转移动作序列$A_1 \cdots A_{2n}$从初始状态转移到达最终状态，最终这棵句法树的分数可以由公式(4-2) 计算，

$$\text{Score}_{\text{joint}}(\mathbf{x},\mathbf{t},\mathbf{d}) = \sum_{A_i=\text{SHIFT(t)}} \mathbf{w}_{\text{pos}} \cdot \mathbf{f}_{\text{pos}}(\text{ST}_i, A_i, t) + \sum \mathbf{w}_{\text{syn}} \cdot \mathbf{f}_{\text{syn}}(\text{ST}_i, A_i) \qquad (4\text{-}2)$$

其中ST表示一个状态，$A$表示动作，$\mathbf{f}$表示特征，$w$表示特征权重，我们使用感知器算法结合提前更新(early-update)算法得到特征权重$w$。基于转移的联合模型中所使用的各种特征如表4-2 所示，其中$w$表示词，$t$表示词性，$begin(\cdot)$和$end(\cdot)$分别表示词$w$第一个字符和最后一个字符，$prefix(w,k)$和$suffix(w,k)$分别表示词$w$长

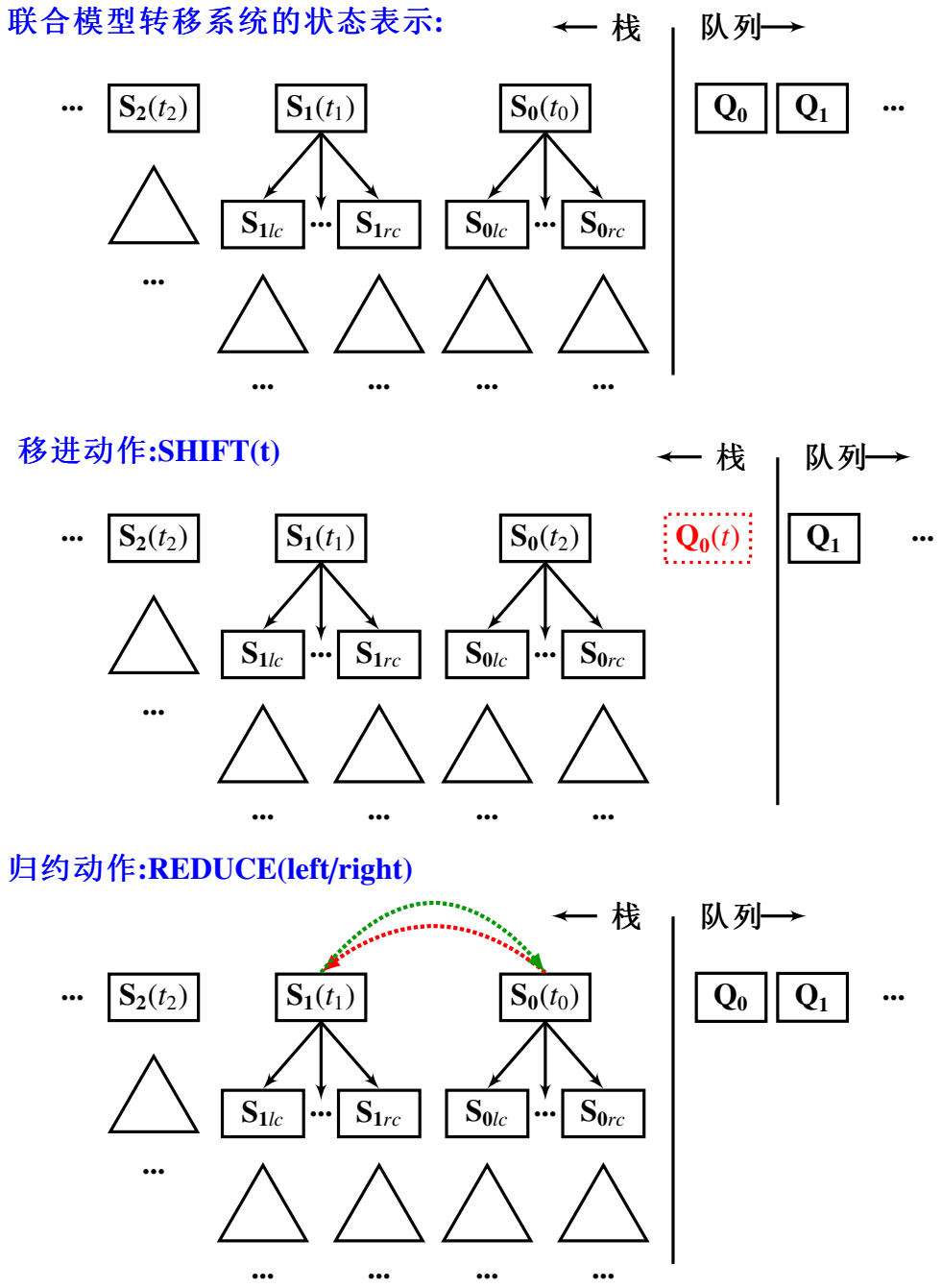



图 4-2 基于转移的联合模型中状态及其相关动作定义。

Fig. 4-2 States and actions of the transition-based joint model.

度为$k$的前缀和后缀，$rval$和$lval$分别表示一个节点的右子树的个数和左子树的个数，$rc$和$lc$分别表示一个节点的最右孩子节点和最左孩子节点，$rc2$和$lc2$分别表示一个节点的最右第二个孩子节点和最左第二个孩子节点，$dist$表示某个依存弧的父亲节点和该节点的距离，#punc$(h,m)$表示词$h$和词$m$之间所包含的标点符号的数目。

表 4-2 基于转移的联合模型的特征模板。

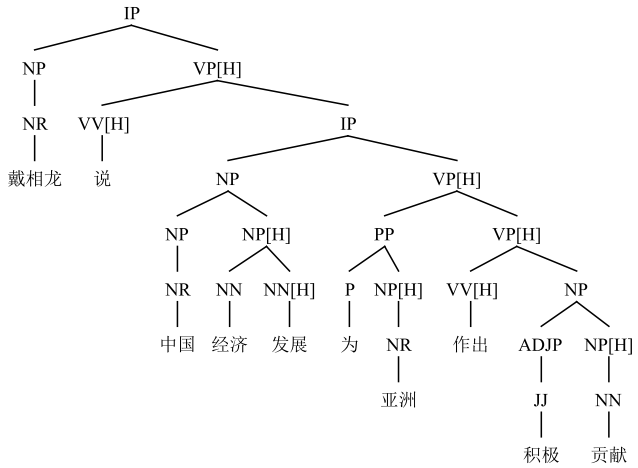
Table 4-2 Feature templates for the transition-based joint model.

| 类别 | 特征模板 |
|---|---|
| 词性标注 | $t_i w_i \ \ t_i t_{i-1} \ \ t_i t_{i-1} t_{i-2} \ \ t_i w_{i+1} \ \ t_i w_{i-1}$ |
| | $t_i w_i \,.\, end(w_{i-1}) \ \ t_i w_i \,.\, begin(w_{i+1}) \ \ t_i \,.\, C_k(w_i)$ |
| | $t_i \,.\, prefix(w_i,k) \ \ t_i \,.\, suffix(w_i,k)$ 其中$1 \le k \le 3$ |
| | $t_i \,.\, S_0 w \ \ t_i \,.\, S_0 t \ \ t_i w_i \,.\, S_0 w \ \ t_i w_i \,.\, S_0 t$ |
| | $t_i w_i \,.\, begin(S_0 w) \ \ t_i w_i \,.\, end(S_0 w)$ |
| | $t_i \,.\, S_0 t \,.\, S_{0rc} t \ \ t_i \,.\, S_0 t \,.\, S_{0lc} t \ \ t_i w_i \,.\, S_0 t \,.\, S_{0rc} t \ \ t_i w_i \,.\, S_0 t \,.\, S_{0lc} t$ |
| 依存句法分析 | $S_0 w \ \ S_0 t \ \ S_0 wt \ \ S_1 w \ \ S_1 t \ \ S_1 wt \ \ Q_0 w \ \ S_0 w \,.\, S_1 w \ \ S_0 t \,.\, S_1 t$ |
| | $S_0 w \,.\, S_1 t \ \ S_0 t \,.\, S_1 w \ \ S_0 wt \,.\, S_1 t \ \ S_0 t \,.\, S_1 wt$ |
| | $S_0 wt \,.\, S_1 w \ \ S_0 w \,.\, S_1 wt \ \ S_0 wt \,.\, S_1 wt$ |
| | $S_0 t \,.\, S_1 t \,.\, S_{1lc} t \ \ S_0 t \,.\, S_1 t \,.\, S_{1rc} t \ \ S_0 w \,.\, S_1 t \,.\, S_{1rc} t \ \ S_0 w \,.\, S_1 t \,.\, S_{1rc} t$ |
| | $S_0 t \,.\, S_1 t \,.\, S_{0lc} t \ \ S_0 t \,.\, S_1 t \,.\, S_{0rc} t \ \ S_0 w \,.\, S_1 t \,.\, S_2 t$ |
| | $S_0 w \,.\, dist \ \ S_0 t \,.\, dist \ \ S_1 w \,.\, dist \ \ S_1 t \,.\, dist$ |
| | $S_0 w \,.\, lval(S0) \ \ S_0 t \,.\, lval(S0) \ \ S_1 w \,.\, lval(S1)$ |
| | $S_1 t \,.\, lval(S1) \ \ S_1 w \,.\, rval(S1) \ \ S_1 t \,.\, rval(S1)$ |
| | $S_{0lc} w \ \ S_{0lc} t \ \ S_{1lc} w \ \ S_{1lc} t \ \ S_{1rc} w \ \ S_{1rc} t$ |
| | $S_{0lc2} w \ \ S_{0lc2} t \ \ S_{1lc} w \ \ S_{1lc2} t \ \ S_{1rc2} w \ \ S_{1rc2} t$ |
| | $S_0 t \,.\, S_{0lc} t \,.\, S_{0lc2} t \ \ S_1 t \,.\, S_{1lc} t \,.\, S_{1lc2} t \ \ S_1 t \,.\, S_{1rc} t \,.\, S_{1rc2} t$ |

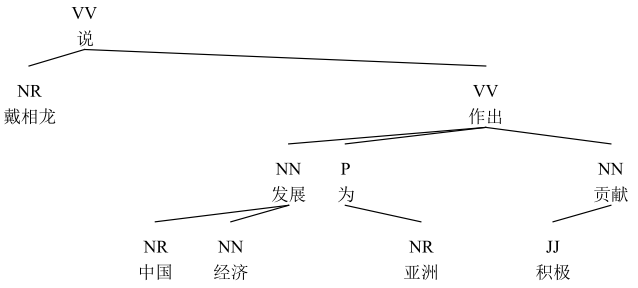
在解码中，基于转移的算法每一步根据当前状态逐步选择动作并生成下一步的状态候选集，进而再往下一步进行搜索。每一步搜索时，都可能存在着很多种动作选择方案，该联合模型采用了柱搜索方法，找出全局最优的转移动作序列。一般情况为了保证和基于图的联合模型在准确率上水平相当，所使用的柱大小为64。

### 4.3.3 基于短语结构的联合模型

基于短语结构的词性标注和依存句法分析的联合模型，首先需要假设对于参与依存句法分析训练的每个句子都有相应的短语结构句法树；然后我们在这个语料上面训练得到一个短语结构句法分析器，从而我们可以通过该短语句法分析器能得到任何句子的短语结构句法树，该短语结构句法分析器必须能不加区分的处理短语节点和词性节点，使得词性标注和短语句法分析能被同时分析出来；最后再将自动分析得到的短语结构句法树根据一些转换规则将其转换成依存结构，便同时得到了词性标注和依存句法分析的结果我们用JConst来表示这种联合模型。




a) 短语结构树，父亲节点已经标注.




b) 依存结构树.

图 4-3 短语结构树与依存结构树对比。

Fig. 4-3 A comparison between constituent and dependency trees

整个过程中有一点至关重要，即这种中间的短语句法结构必须能转换成依存结构。幸运的是我们所使用的句法依存树是根据中文宾州短语句法树库通

过父亲节点发现规则(Head-Finding Rules)直接转换过来的，因此我们的依存句法语料满足这一点。实际上仍然存在有些依存句法树库不是通过短语句法的转换得到，但是我们可以为这些依存树自动构造一棵伪短语句法树，孙薇薇等人在2013年提出了一种自动构建的方法[42]，具体细节可以参考他们的论文。图4-3中显示了中文宾州短语结构句法树和相应的依存结构句法树的一个例子，其中标记为[H]的节点是短语结构树中某个短语的父亲节点，我们可以根据这些父亲节点来自动得到最终的依存结构句法树。

在本章中，我们使用Berkeley parser进行短语句法分析[30]，其主要原因在于Berkeley parser是一种非词汇化的短语句法分析器，即在进行句法分析的时候，没有用到任何词汇相关的特征。这种非词汇化的分析方法和前面两种词汇化的依存句法分析模型本质上有着巨大的区别，从而在进一步融合时，这些差异会带来更好的融合模型性能。

## 4.4 融合模型

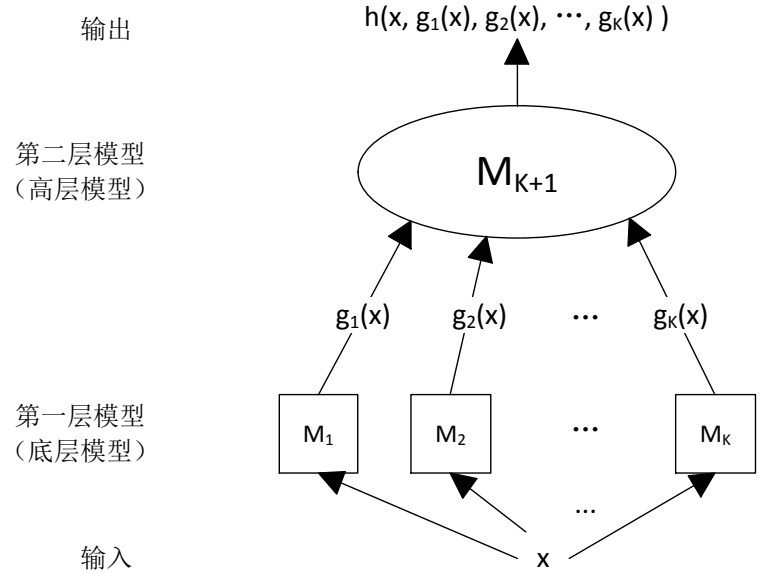



图 4-4 基于栈学习融合模型的核心思想示意图。

Fig. 4-4 The main idea of stacked learning.

系统融合是一种非常自然的提高模型性能的方法。在本章中，我们使用基于栈学习的方法来融合上面三种基本模型[107, 116]，这种方法不仅融合形式优美，而且不会局限于单一模型的输出结果。基于栈学习的模型融合方法在自然语言处理的模型融合上已经有了不少成功的例子，例如分词的融合[60]，词性标注的

融合[117]，以及依存句法的融合[99, 109, 110, 112]，本章我们将其应用在词性和句法的联合模型上。

基于栈学习模型融合的核心思想如图4-4所示，它表现为一个两层的学习器，其中第一层包括1个或者多个模型，用$g_1, \cdots, g_K (K \geq 1) : R^d \rightarrow R$来表示；而第二层只有一个模型，$h : R^{d+K} \rightarrow R$。每个第1层的模型$g_k$都能为输入的$\mathbf{x} \in R^d$自动预测出一个结果$g_k(\mathbf{x})$，第二层的模型将输入$\mathbf{x}$和第一层$K$个模型的结果结合在一起构成输入$\langle \mathbf{x}, g_1(\mathbf{x}), \cdots, g_K(\mathbf{x}) \rangle$，然后根据这个输入做最终的预测$h(\mathbf{x}, g_1(\mathbf{x}), \cdots, g_K(\mathbf{x}))$.

当基于栈的学习方法应用到词性标注依存句法联合模型的融合时，由于存在三种不同的基本联合模型，因此需要确定选取哪种基本模型作为第二层的分析器。对此我们有两种选择，第一种是使用基于图的联合模型作为第二层分析器，另外一种是使用基于转移的联合模型作为第二层分析器。对于基于短语句法结构的联合模型，由于它是一个生成模型，很难融入额外的第一层分析器所提供的信息，所以它不合适作为第二层模型。无论用哪种模型作为第二层的分析器进行融合，其解码和训练算法是不会变化的，变化的只有特征。在使用基于图的或者基于转移的联合模型作为第二层分析器时，由于融入了第一层分析器的结果，因此会加入一些新的特征，具体特征模板如表4-3所示。在表中，上半部分是使用基于图的联合模型作为第二层分析器时所加入的新特征，而下半部分是使用基于转移的联合模型作为第二层分析器时所加入的新特征。

## 4.5 过训练

通过模型融合，我们可以提升联合模型的性能，但是模型融合却使得解码的速度降低了很多。本节我们采用一种过训练的方法来提升联合模型的速度，其本质是利用大规模的未标注数据来使得一个简单快速模型的性能得到加强。过训练最早由Slav Petrov等人在2010年提出[111]，它假设一个任务存在两种不同的模型$M_1$和$M_2$，同时还有大规模未标注数据，其中$M_1$速度非常慢但是准确率高，而$M_2$速度非常快但是准确率却低了不少，过训练方法使用$M_1$去自动解析大规模的未标注数据，然后用自动解析得到的数据加入到$M_2$的训练语料中来进一步训练模型$M_2$，从而使得$M_2$的性能大幅度提升，因此得到了一个速度快而且性能高的模型。

对于词性标注和依存句法联合模型，我们存在一个高精度但是速度慢（速度为平均1句每秒）的融合联合模型JGraph(JTrans, JConst)或

表 4-3 基于栈学习的融合模型所加入的新特征。

Table 4-3 Feature templates for the integrated models via stacked learning.

| 类别 | 特征模板 |
|---|---|
| 基于图的联合模型作为第二层分析器 | |
| 词性标注 | $\{\hat{t}_m^{\text{JTrans}}, \hat{t}_m^{\text{JTrans}} \circ \hat{t}_{m-1}^{\text{JTrans}}, \hat{t}_m^{\text{JTrans}} \circ \hat{t}_{m+1}^{\text{JTrans}}, \hat{t}_{m-1}^{\text{JTrans}} \circ \hat{t}_{m+1}^{\text{JTrans}}\} \otimes \{t_m, w_m \circ t_m\}$<br>$\{\hat{t}_m^{\text{JConst}}, \hat{t}_m^{\text{JConst}} \circ \hat{t}_{m-1}^{\text{JConst}}, \hat{t}_m^{\text{JConst}} \circ \hat{t}_{m+1}^{\text{JConst}}, \hat{t}_{m-1}^{\text{JConst}} \circ \hat{t}_{m+1}^{\text{JConst}}\} \otimes \{t_m, w_m \circ t_m\}$<br>{Whether $\hat{t}_m^{\text{JTrans}}$ is identical to $\hat{t}_m^{\text{JConst}}$?} $\otimes\{\hat{t}_m^{\text{JTrans}} \circ t_m, \hat{t}_m^{\text{JTrans}} \circ w_m \circ t_m\}$ |
| 依存句法分析 | {Whether $h \curvearrowright m$ is in $\hat{\mathbf{d}}^{\text{JTrans}}$?} $\otimes\{t_h, t_m, t_h \circ t_m\}$<br>{Whether $h \curvearrowright m$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?} $\otimes\{t_h, t_m, t_h \circ t_m\}$<br>{Whether the heads of $m$ are identical in $\hat{\mathbf{d}}^{\text{JTrans}}$ and $\hat{\mathbf{d}}^{\text{JConst}}$?} $\otimes$ {Whether $h \curvearrowright m$ is in $\hat{\mathbf{d}}^{\text{JTrans}}$?} $\otimes\{t_h, t_m, t_h \circ t_m\}$<br>{Whether $h \curvearrowright m$ and $h \curvearrowright s$ are in $\hat{\mathbf{d}}^{\text{JTrans}}$?}$\otimes\{t_h, t_m, t_h \circ t_m\}$<br>{Whether $h \curvearrowright m$ and $h \curvearrowright s$ are in $\hat{\mathbf{d}}^{\text{JConst}}$?}$\otimes\{t_h, t_m, t_h \circ t_m\}$<br>{Whether $g \curvearrowright h \curvearrowright m$ is in $\hat{\mathbf{d}}^{\text{JTrans}}$?}$\otimes\{t_h, t_m, t_h \circ t_m\}$<br>{Whether $g \curvearrowright h \curvearrowright m$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?}$\otimes\{t_h, t_m, t_h \circ t_m\}$ |
| 基于转移的联合模型作为第二层分析器 | |
| 词性标注 | $\{t_m, w_m \circ t_m\} \otimes \{\hat{t}_m^{\text{JGraph}}, \hat{t}_m^{\text{JGraph}} \circ \hat{t}_{m-1}^{\text{JGraph}}, \hat{t}_m^{\text{JGraph}} \circ \hat{t}_{m+1}^{\text{JGraph}}, \hat{t}_{m-1}^{\text{JGraph}} \circ \hat{t}_{m+1}^{\text{JGraph}}\}$<br>$\{t_m, w_m \circ t_m\} \otimes \{\hat{t}_m^{\text{JConst}}, \hat{t}_m^{\text{JConst}} \circ \hat{t}_{m-1}^{\text{JConst}}, \hat{t}_m^{\text{JConst}} \circ \hat{t}_{m+1}^{\text{JConst}}, \hat{t}_{m-1}^{\text{JConst}} \circ \hat{t}_{m+1}^{\text{JConst}}\}$<br>{Whether $\hat{t}_m^{\text{JGraph}}$ is identical to $\hat{t}_m^{\text{JConst}}$?} $\otimes\{\hat{t}_m^{\text{JGraph}} \circ t_m, \hat{t}_m^{\text{JGraph}} \circ w_m \circ t_m\}$ |
| 依存句法分析 | {Whether $s_0 \curvearrowleft s_1$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?, Whether $s_0 \curvearrowright s_1$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?} $\otimes\{s_0.t, s_1.t, s_0.t \circ s_1.t\}$,<br>{Whether $s_0 \curvearrowleft s_1$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?, Whether $s_0 \curvearrowright s_1$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?} $\otimes$ {Whether $s_0 \curvearrowleft (s_{0lc})$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?, Whether $s_0 \curvearrowleft (s_{0rc})$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?, Whether $s_1 \curvearrowleft (s_{1lc})$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?, Whether $s_1 \curvearrowleft (s_{1rc})$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?} $\otimes\{s_0.t, s_1.t, s_0.t \circ s_1.t\}$<br>{Whether $s_0 \curvearrowleft s_1$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?, Whether $s_0 \curvearrowright s_1$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?} $\otimes\{s_0.t, s_1.t, s_0.t \circ s_1.t\}$,<br>{Whether $s_0 \curvearrowleft s_1$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?, Whether $s_0 \curvearrowright s_1$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?} $\otimes$ {Whether $s_0 \curvearrowleft (s_{0lc})$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?, Whether $s_0 \curvearrowleft (s_{0rc})$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?, Whether $s_1 \curvearrowleft (s_{1lc})$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?, Whether $s_1 \curvearrowleft (s_{1rc})$ is in $\hat{\mathbf{d}}^{\text{JConst}}$?} $\otimes\{s_0.t, s_1.t, s_0.t \circ s_1.t\}$<br>{Whether the heads of $s_0$ are identical in $\hat{\mathbf{d}}^{\text{JGraph}}$ and $\hat{\mathbf{d}}^{\text{JConst}}$?, Whether the heads of $s_1$ are identical in $\hat{\mathbf{d}}^{\text{JGraph}}$ and $\hat{\mathbf{d}}^{\text{JConst}}$?} $\otimes$ {Whether $s_0 \curvearrowleft s_1$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?, Whether $s_0 \curvearrowright s_1$ is in $\hat{\mathbf{d}}^{\text{JGraph}}$?} $\otimes\{s_0.t, s_1.t, s_0.t \circ s_1.t\}$ |

者JTrans(JGraph, JConst)，因此我们只需要设计一个精度低但是速度非常快的简单联合模型，便可以得到一个高效率高性能的词性标注依存句法联合模型，这两个模型分别对应于上面提到的$M_1$和$M_2$。

我们通过改变基于转移的联合模型柱搜索算法中柱的大小来实现我们的简单联合模型。柱为64时是前面提到的一个基准模型，当柱大小逐渐降低时，模型速度就会变快，但是如果仍然使用原来的训练语料，则模型性能就会显著下降。因此我们采用前面提出的最好的融合联合模型自动解析100万句规模的原始句子，然后加入到柱大小降低后的简单联合模型的训练语料中，使得简单联合模型的性能大大提升，甚至超过了基准的64柱大小的联合模型的性能，而且速度也由原来的9句每秒到达最终的120句每秒。

## 4.6 实验

我们在中文宾州树库(The Penn Chinese Treebank, CTB)5.1版上进行实验来验证我们提出的方法。CTB5.1数据统计信息如表4-4所示，我们采用标准的划分方法将这一数据集划分成为训练、开发以及测试集。中文宾州树库是一个短语句法树库，我们通过张岳等人2008年提出的规则[38]，将中文宾州树库短语结构句法树转换成依存结构。在评价词性标注性能时，我们使用词性标注准确率，即词性标注正确的词的总数占所有词比例；在评价依存分析性能时，我们使用不带标签的依存弧准确率(Unlabeled Attachment Score, UAS)，即父亲节点被正确找到的词的个数占所有词的比例，另外还使用了根节点识别准确率(Root Accuracy, RA)以及整个句子正确识别准确率(Completely Match, CM)，在评价依存时，我们忽略了标点符号。

表 4-4 语料统计信息.

Table 4-4 Corpus statistics.

| | 划分方法 | 句子数目 | 词数目 |
|---|---|---|---|
| Training | 001–815; 1001–1136 | 16,118 | 437,859 |
| Dev | 886–931; 1148–1151 | 804 | 20,453 |
| Test | 816–885; 1137–1147 | 1,915 | 50,319 |

### 4.6.1 基准系统性能

首先，我们给出了三个基准联合模型的性能，对于这三种联合模型，还分别与各自的串行模型做了对比。最终结果如表4-5所示，其中以P开头的为相应的串行模型，里面Li-12和Joint-ZN$^{-}$分别为李正华2012年以及Jun Hatori在2011年同样的方法论文上的结果[61, 64]，里面特征选取有稍微的不一样。从表中可以看出我们所使用的三个基本模型，和串行模型相比，在句法上都有一定性能的提升，基于短语结构的联合模型JConst取得了最好的句法分析结果，但是在词性上面，这一模型的性能却最差，而基于图的联合模型在词性标注上面取得了最好的效果。

表 4-5 词性标注依存句法基准模型性能。

Table 4-5 The performances of baseline joint models.

| | 句法 | | | 词性标注 |
|---|---|---|---|---|
| | UAS | RA | CM | |
| 基本联合模型 | | | | |
| JGraph | 80.88 | 75.55 | 28.83 | **94.51** |
| JTrans | 80.98 | 75.54 | **29.68** | 94.21 |
| JConst | **81.03** | **78.12** | 28.01 | 93.45 |
| 串行模型 | | | | |
| PGraph | 79.52 | 75.34 | 26.70 | 94.11 |
| PTrans | 79.30 | 75.73 | 27.80 | |

从上面显示的结果我们可以大致看出这三个基本联合模型有着比较大的区别，这里我们进一步针对每个具体的句子来观察这三个基准模型的性能，了解这三个模型的错误分布情况。我们分别计算三个基准联合模型在同一个句子上依存分析的性能，然后用散点图做了对比。散点图中的每个点，其横坐标表示其中一个模型的准确率，而纵坐标表示另外一个联合模型的准确率，最终比较结果如图4-5所示。从图中可以看到，散点图结果分布均匀，绝大部分点落在了对角线外，这一结果表明了这三个联合模型虽然总体性能相差不大，但是它们的具体错误分布差别是比较大的，因此模型融合将会是比较有效的提升系统准确率的手段。

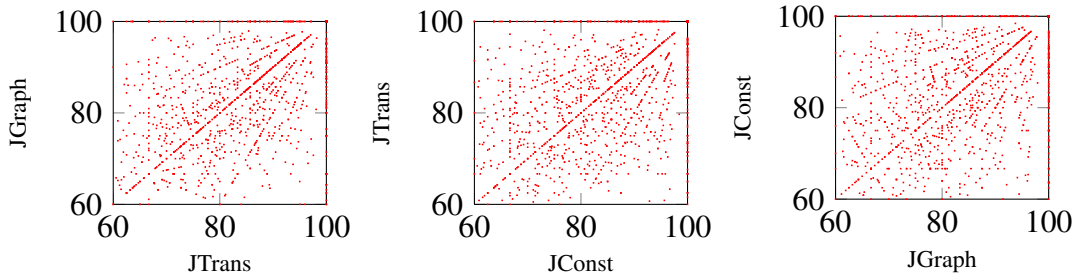



图 4-5 基准联合模型依存分析性能散点图对比。

Fig. 4-5 The scatter figures for dependency accuracies of the baseline joint models.

### 4.6.2 融合模型性能

前面我们给出了三个基准联合模型的性能，并且说明了这三个基准模型虽然性能相似，但是错误分布差别比较大，模型融合将会取得比较好的效果。在这里，我们将逐步给出融合联合模型的实验结果。首先介绍以基于图的联合模型为融合模型第二层的结果，此时基于第一层的模型可以为基于转移的联合模型、基于短语句法结构的联合模型，以及这两个联合模型一起都作为第一层模型，其结果如表4-6上半部分所示。对于采用基于转移的联合模型为第二层模型的融合模型，我们也做了类似的处理，其结果如表4-6下半部分所示。从实验结果中可以发现，使用基于转移的联合模型为融合模型第二层时，性能会更好一点。最终使用基于转移的联合模型为融合模型第二层并且使用基于图的以及基于短语结构的联合模型一起作为融合模型第一层时取得最好的结果，这样得到的模型相当于三个模型一起进行融合，和单独的基准联合模型相比，在句法分析上的性能提升接近3%，而在词性标注上的性能提升至少为0.4%。

### 4.6.3 联合模型实验结果分析

融合联合模型除了能提高词性标注和依存分析性能之外，还可以让我们更加全面的分析中文词性标注与依存句法之间的关系。本节中主要将三个基准联合模型以及一个性能最好的融合联合模型和串行模型进行比较，来观察和比较词性和依存句法之间的相互联系。首先，我们研究联合模型对词性标注的影响，第一种方案通过比较父亲节点被正确找到以及父亲节点被错误分析的词，观察这两类词的词性标注准确率，这样便可以知道依存句法对词性标注的影响。表4-7给出了相应的分析结果，很明显，如果该词的依存父亲节点正确找到

表 4-6 词性标注依存句法融合模型性能。

Table 4-6 The performances of integrated joint models.

| | 句法 | | | 词性标注 |
|---|---|---|---|---|
| | UAS | RA | CM | |
| JGraph(JTrans, JConst) | 83.59 | 80.79 | 31.47 | 94.91 |
| JGraph(JConst) | 83.01 | 79.69 | 31.34 | 94.72 |
| JGraph(JTrans) | 82.04 | 78.17 | 30.21 | 94.52 |
| JGraph | 80.88 | 75.55 | 28.83 | 94.51 |
| JTrans(JGraph, JConst) | **83.98** | **81.29** | **32.15** | **94.95** |
| JTrans(JConst) | 83.23 | 80.73 | 31.55 | 94.44 |
| JTrans(JGraph) | 82.22 | 78.03 | 30.58 | 94.75 |
| JTrans | 80.98 | 75.54 | 29.68 | 94.21 |

的话，词性标注的准确率会比父亲节点错误的情况高出10%以上，联合模型更容易夸大这种差距，因为联合模型中句法分析错了，会直接影响到词性标注的结果。

表 4-7 依存句法父亲节点对词性标注的影响。

Table 4-7 The influences of dependency heads on POS-tagging accuracies.

| | **PGraph** | **JGraph** | **JTrans** | **JConst** | **JTrans(JGraph, JConst)** |
|---|---|---|---|---|---|
| 父亲节点正确 | 96.23 | 96.92↑ | 96.64↑ | 96.36↑ | 97.03↑ |
| 父亲节点错误 | 86.65 | 85.34↓ | 84.16↓ | 81.79↓ | 84.7↓ |

第二方案，我们观察联合模型中哪类词性错误会被进一步扩大，以了解句法对词性错误类别的影响。最终的分析结果如图4-6所示，图中的结果显示了联合模型和串行模型相比时，各个具体词性对错误率减少百分比，在横轴上面的表示联合模型能使得这些错误减少，而在横轴下面的表示联合模型能使得这些错误增加。和串行模型相比，联合模型能很好的区分名词被分析成动词(NN → VV)的情况以及“的”字词性的区分(DEC → DEG和DEG → DEC等等)，但是却很难区分专有名词被分析成普通名词NR → NN 和普通名词被分析成形容词NN → JJ等情况，这些容易区分的词性，正好体现了这些词在句法分析中的重要性；相反例如普通名词NN和专有名词NR的区别在句法中不明显，或者说它们可以完全充当相似的句法角色，因此联合模型也很难区别这些词性

错误模式。

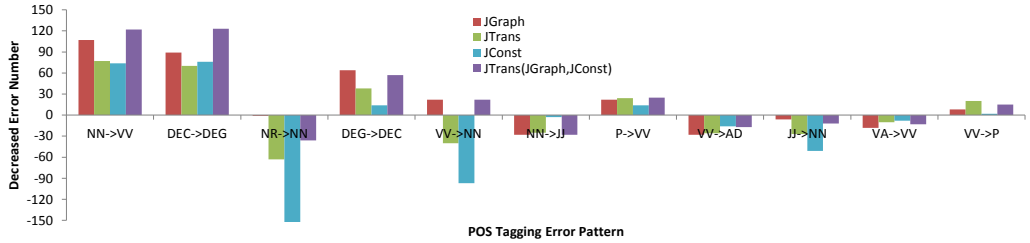



图 4-6 对比串行模型，联合模型词性标注错误模式分析。

Fig. 4-6 The POS-tagging error patterns of the joint and pipeline models.

另外值得一提的是，我们发现某个词的父亲节点在该词的左边还是该词的右边也对词性标注性能有一定影响，具体结果如表4-8所示。父亲节点在右边的词（左弧），往往在联合模型中没有明显的提升效果，因为这些词大部分是修饰词，例如普通名词NN，形容词JJ，副词AD，长距离依存比较少，串行模型中的线性模型即可区分这些，而且它们的正确区分对句法帮助意义不大；而父亲节点在左边的词（右弧），大部分是动词VV之类的，在联合模型中有很明显的提升效果。

表 4-8 父亲节点在左边还是右边对词性标注准确率的影响。

Table 4-8 The influences of dependency head direction on POS-tagging.

| | **PGraph** | **JGraph** | **JTrans** | **JConst** | **JTrans(JGraph, JConst)** |
|---|---|---|---|---|---|
| $w_i \curvearrowright w_j$ | 94.12 | 94.24 | 93.78 | 92.86 | 94.51 |
| $w_i \curvearrowleft w_j$ | 93.95 | 94.92 | 94.54 | 94.54 | 95.62 |

前面的分析主要针对联合模型对词性标注性能的影响，在这里我们进一步分析联合模型对句法分析的影响。第一种方案，假设词性分析的结果不正确，观察此时依存分析的性能受到多大的影响，图4-7给出了分析结果。我们发现，词性错误模式对句法影响比较大包括NN → VV，以及DEC → DEG等等，这些错误会使得联合模型中句法的性能会大幅度下降，这表明了这些词性的区分对句法至关重要，因此在联合模型中，这两个词性更容易被正确区分。而对于有些词性错误模式，对句法的性能影响非常小，例如NR → NN以及JJ → NN影响句法的，这表明这两个词性的正确区分对句法并不怎么影响，因此在联合模型中，这两个词性往往会被混淆。这一分析结果也和联合模型对词性错误模式的分析结果相符合。

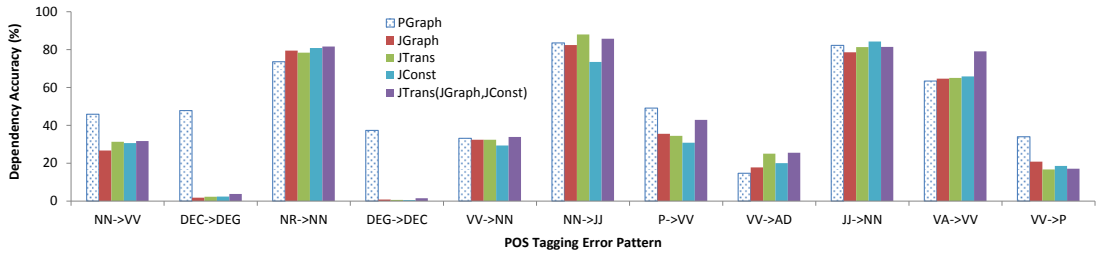



图 4-7 词性标注错误模式对依存分析性能的影响

Fig. 4-7 Dependency accuracies of different POS-tagging error patterns.

第二步，我们观察左弧和右弧的性能在联合模型中的变化。因为左弧大部分是修饰词，也大部分是局部依存弧，前面我们也看到，这类弧词性上面也没有明显的提升，因此联合模型对左弧影响应该不大；但是由于联合模型能够提升右弧词的词性标注性能，因此我们可以预见右弧的句法分析性能在联合模型中应该会有较大的提升。最终实验分析结果如表4-9所示，从表中数据能看到，单一联合模型右弧性能的提升是比较明显的，而左弧性能的变化相对右弧来说略小一点，当然融合联合模型是对无论左弧还是右弧都有比较大的提升，而且表中数据也确实表明，从总体来看，左弧要比右弧性能高，容易分析。

表 4-9 弧方向对依存分析的影响。

Table 4-9 The influences of head direction on dependency accuracies.

| | **PGraph** | **JGraph** | **JTrans** | **JConst** | **JTrans(JGraph, JConst)** |
|---|---|---|---|---|---|
| $w_i \curvearrowright w_j$ | 81.03 | 82.01 | 82.44 | 81.75 | 84.81 |
| $w_i \curvearrowleft w_j$ | 76.45 | 78.92 | 77.73 | 79.7 | 82.35 |

## 4.6.4 过训练

最后我们介绍过训练的实验结果，和第4.5节中提到一样，我们通过改变基于转移的联合模型的柱大小，来使得联合模型的速度变快。为了弥补柱大小的降低所带来的性能损失，我们根据过训练的原理，为柱大小减小后的联合模型增加了大量的训练语料。在我们的实验中，我们从LDC中文Gigaword语料（LDC2007T03）中随机选择了100万句语料，去掉了其中过长的句子，并利用张岳和Clark在2007年提出的方法进行自动分词[13]，其中的分词模型也是利用CTB5.1训练出来的，然后利用前面性能最好的融合模型JTrans(JGraph, JConst)自动解析该句子，得到这些句子的自动词性和依存句法分析结果。最终我们将

这些自动解析的语料分别加入到柱大小为64，32，16，8，4，2和1的基于转移的联合模型中，与原始的CTB51训练语料合并在一起进行模型训练。

表4-10显示了最终的结果，我们依次给出了柱大小为64，32，16，8，4，2和1的联合模型的性能以及速度，同时也给出了不加这些自动训练语料时的联合性能以及速度。从最终结果中我们可以看出，无论是使用过训练还是未使用过训练，从柱大小为4降到柱大小为2的时候，性能都有一个比较明显的下降。当使用过训练之后，柱大小为2的性能也超过了不使用过训练时柱大小为64的联合模型，但是其速度由原来的8.9句每秒上升到了现在的120.7句每秒，提高了10倍多。从表中我们也能看到，同样的柱大小，使用过训练后，速度会有一定的下降，这是由于特征数目的增加导致的，通过观察模型大小，我们模型变大了十倍左右，也就是特征数目增加了十倍左右，这样会在哈希查找特征权重时带来比较大的开销，因此导致了速度的下降。

## 4.7 本章小结

本章我们提出了一种速度快而且性能高的词性标注和依存句法的联合模型，具体方法是通过模型融合与过训练相结合的方法来实现的。这一方法具有比较好的通用性，可以应用在其它任务的联合模型上面。

我们介绍了基于栈学习的融合联合模型，其中基准的词性标注依存句法联合模型包括基于图的联合模型、基于转移的联合模型和基于短语句法结构的联合模型。通过基于栈学习的方式进行模型融合，最终可以将依存句法的性能提升接近3%，而且我们发现采用基于转移的联合模型作为栈学习融合模型的第二层时能取得更好的性能。

融合联合模型虽然取得了最好的性能，但是其速度也是最慢的而且难以忍受的，而且即便是三个基准的联合模型中速度最快也达不到10句每秒，因此我们进一步采用了基于过训练的方法来提升一个速度快的简单联合模型的性能。由于基于转移的联合模型可以非常方便的通过调整柱大小来平衡联合模型的准确率和速度，因此我们以它为基础，并结合过训练方法，实现了一个速度快而且性能好的联合模型。最终我们提出的联合模型达到了120句每秒的分析速度，而且性能也和原有的基准联合模型性能相当。

表 4-10 过训练实验的最终联合模型性能。

Table 4-10 The performances of up-training models.

| 柱大小 | 句法 | | | 词性 | 速度 |
|---|---|---|---|---|---|
| | UAS | RA | CM. | 标注 | 句/秒 |
| 过训练 | | | | | |
| 64 | 83.07 | 78.64 | 31.98 | 94.70 | 5.4 |
| 32 | 82.85 | 77.59 | 31.78 | 94.80 | 11.8 |
| 16 | 82.83 | 77.53 | 31.83 | 94.86 | 21.4 |
| 8 | 82.52 | 77.17 | 31.57 | 94.84 | 37.7 |
| 4 | 82.31 | 76.49 | 31.15 | 94.88 | 67.8 |
| 2 | 81.18 | 75.50 | 30.52 | 94.64 | 120.7 |
| 1 | 78.51 | 71.31 | 27.33 | 93.98 | 290.3 |
| 未使用过训练 | | | | | |
| 64 | 80.98 | 75.54 | 29.68 | 94.21 | 8.9 |
| 32 | 80.54 | 74.55 | 29.37 | 94.16 | 17.2 |
| 16 | 80.72 | 74.86 | 29.94 | 94.22 | 32.6 |
| 8 | 80.50 | 74.29 | 29.26 | 94.24 | 60.8 |
| 4 | 80.02 | 72.77 | 28.27 | 94.14 | 104.9 |
| 2 | 78.41 | 70.83 | 28.11 | 93.79 | 200.1 |
| 1 | 76.01 | 68.01 | 24.81 | 93.08 | 380.6 |

# 第 5 章 词典和句子标注相结合的分词词性联合模型领域自适应

## 5.1 引言

领域自适应是目前自然语言处理一个比较热门的研究点，也是非常难解决的一个问题。但是对于联合模型，很少有专门针对领域自适应进行相关的研究工作。在本章，我们以分词词性标注的联合模型为例，来讨论这个问题。过去的对中文分词词性标注的研究，大部分都建立在固定的评测数据集上，例如中文宾州树库（Chinese Penn Treebank，CTB）上面，这种评测数据集的训练、开发以及测试语料都处于同一领域，这样词的词性分布概率以及词性之间的转移分布概率在这三个集合上面都非常一致，最终的测试集性能也达到了94%。而在实际应用中，面临的句子可能会是一个完全不一样的领域，例如一部小说中的句子，或者是微博数据。在这种情况下，词的词性分布概率以及词性之间的转移分布概率便会完全不一样，从而导致最终的性能可能会有比较严重的下降。我们真实的实验也验证了这一点，当前在中文宾州树库语料上性能最好的分词词性联合模型应用在网络小说上时，性能会下降至82%左右[118]。

对于领域迁移问题，有两种比较可取的思想，第一种是无指导和有指导方法相结合的半指导的思想。由于无指导的方法不需要训练语料，不局限于某个特定的领域，如果我们能在有指导的模型中融入无指导的方法所学习出来的领域知识，那么会使得该模型的领域自适应有一定的增强。过去，无指导的分词词性标注研究主要体现在串行模型上面，也就是分别在这两个任务下进行。无指导的分词主要采用两个字符串的点互信息(Point Mutual Information, PMI) 来判断它们是否应该连接在一起，或者使用一段字符串的左右两边的字符变化信息来判断这个字符串是否为一个独立的分词片段[119]。对于无指导的词性标注，大多采用隐马尔科夫模型结合期望最大算法，从大规模无标注数据中估计词性与词性之间的转移概率和词性到词之间的发射概率，但是在采用期望最大算法训练参数时往往需要指定一个初始的转移概率和发射概率，在给定这个初始概率之后，最终模型会在一个局部最优点收敛。不少研究者发现，如果这个初始值设置的比较好，无指导的算法会达到和有指导的算法非常接近的水平。很多研究者对这个初始值的设置做了不少尝试，最有效的方法是采用比较合适

的词典，结合标签传播算法，来获取这个初始值，也有部分研究者通过标注少量句子级数据来进行初始化的[120]。对于分词词性标注联合模型的领域自适应，Yang Liu和张岳尝试使用自学习(Self-Training)和聚类(Clustering)来提升不同目标领域的性能，提升效果在2%左右[118]。

第二种比较可取的方法是通过标注少量的领域自适应数据来解决领域迁移问题。单纯的第一种方法，如果不使用少量领域标注数据，性能能在原有的基础上有2%的提高也需要花费很大的精力，例如Yang Liu和张岳所使用的方法使得性能提升了2%[118]。在前面也提到，无指导的词性标注模型如果能配合少量的标注数据（词典或者句子）便可以使得词性标注的性能大大提升。实际上词典标注或者句子标注对于分词也同样适用，分词词典可以使用词性标注中所用到的词典。这种标注少量目标领域数据的方法来解决领域自适应问题，也有不少相关的研究工作。一方面针对句子级标注的领域自适应，Hal DaumeIII在2007 年提出了一种方法使得目标领域的性能在有限的少量标注下提升最大[121]。另一方面部分研究者意识到句子级语料标注的标注工作量一般要远远高于词典标注，因此他们建议了使用词典标注代替句子标注。最具有代表性的是Dan Garrette等人提出一个方案，他们通过两个小时的词典标注构建一个实际的词性标注器[122, 123]，虽然他们将该方法应用在资源短缺的稀有语言上面，但是这一方法对领域迁移有很好的指导作用。

在本章中，我们扩展Dan Garrette等人的方法，将他们的工作应用在了领域自适应的场景上。首先同他们的工作类似，我们比较词典标注和句子标注对中文分词词性标注联合模型的影响，进一步我们提出了一种词典标注和句子标注相结合的方法。实验表明，这种结合的标注方法更能充分的利用标注的人力资源，在相同的标注代价下，词典标注和句子标注相结合的方法更能有效的增强分词词性标注联合模型的领域自适应能力。

## 5.2 相关工作

我们的分词词性联合模型采用的是张岳等人在2010年提出的模型，这一模型简单快速，属于判别模型，能方便的融入各种复杂特征[3]。我们对这一模型进行扩展，使它能利用大量标注的词典特征，从而能将词典的标注信息融入模型中。当然在分词词性标注的联合模型方面，也有不少其它的方法，例如姜文斌等人基于重排序的模型[1]，以及孙薇薇的基于栈学习的模型等等[60]，我们的方法同样也能应用在这些模型上面。

使用词典标注的方法来提升词性标注器的性能已经有了不少的研究工作。在英语方面，相Garrette和Baldridge在2012年提出了利用标签传播算法（Label Propagation，LP）改进了词性标注的隐马尔可夫模型[120]；对于其它小众语种，这方面的工作也非常多[122–124]。这些方法都假定一定规模的已标注词典已经存在（有些工作甚至是直接从测试语料中直接抽取这样的词典），然后利用这一词典去提升某个语言词性标注器的性能。我们的研究工作主要考虑的是领域迁移问题，而且更注重中文的特性。由于在中文的原始句子中没有明显的词语界限，因此我们要考虑的是中文分词词性标注的联合模型。

关于使用句子标注这一方面的工作也有不少。Yang Liu和张岳使用了自学习的方法自动标注目标领域句子的分词词性结果[118]，在此基础上重新训练建立一个目标领域的分词词性标注器。当然使用这种自动的方法获取句子标注所带来的性能提升是很有限的，一个复杂的模型能带来2%的提升已经非常不错了，因此我们主张采用几个小时的人工标注来解决这一问题。如何去使用有限的人工标注使得模型在目标领域的性能达到最好也是一个非常值得研究的问题，不少研究者对其进行了展开研究，例如Hal DaumeIII 在2007年提出将领域相关特征和领域无关特征分开[121]，然后使用源领域语料和目标领域语料相结合来训练这两类特征的权重。我们所采用的思想与之类似，但是我们不希望针对每个领域训练一个模型，我们更希望使用同样的特征，这些特征和目标领域标注资源相结合时我们的模型便能直接捕捉领域信息，从而使得模型在目标领域上的效果大大提升。

关于同时使用句子标注和词典标注的工作非常少，部分研究者采用了自动标注的方法，例如Wang等人在IJCNLP 2011年的工作[77]，他们的工作重点放在了提升同一领域的性能上面，而且他们针对的是串行模型。我们的研究重点在于领域迁移时，分词词性标注联合模型的的性能。

## 5.3 领域自适应问题概述

领域自适应是统计自然语言处理领域一个非常重要的问题，产生这一问题的根源是因为自然语言处理任务中的某个统计模型的训练语料往往属于某个特定的领域，而这一特定领域和实际应用领域中的文本差异很大，因此导致了该模型在面向实际领域中的文本时性能急剧的下降。具体以分词词性标注的联合模型为例，一般情况下我们的分词词性标注联合模型通常是利用中文宾州树库所属的新闻领域数据训练而得到的，我们称该领域为源领域或训练领域，而从

这一联合模型应用到实际的句子中时（我们称实际句子所属的领域为目标领域或者应用领域），如果源领域和目标领域中字、词、词性以及句子的分布完全不一样时，便会导致该分词词性标注的联合模型的性能严重下降，有些情况下甚至会下降10%以上。

在本章中，我们主要关注两种针对分词词性标注联合模型而提出的领域自适应方法：基于句子标注的领域自适应和基于词典标注的领域自适应。首先我们定义一些符号以方便后续的形式化描述：我们用$C_s$表示源领域标注句子的集合，$C_t$ 表示目标领域标注句子的集合，然后$\Omega_t$表示源领域标注的词典，这个可以直接从源领域标注句子集合中直接提取，$\Omega_t$表示目标领域标注的词典。基于上面的定义，基于句子标注（*Sentence-Supervised*或者*Token-Supervised*）的领域自适应是使用$C_s$和$C_t$来提升目标领域的分词词性分析性能，而基于词典标注（*Lexicon-Supervised*或者*Type-supervised*）的领域自适应是使用$C_s$ 和$\Omega_t$来提升目标领域的分词词性分析性能。

## 5.4 主要方法

本章我们介绍一种词典标注和句子标注相结合的方法来解决领域自适应所带来的性能急剧下降问题，这一方法实际上是对Dan Garrette 等人提出方法的一个扩充[122]。首先我们从目的上进行扩充，从原来的跨语言研究（利用资源丰富语言的词性标注器来帮助资源稀缺语言的词性标注），转变为本章的跨领域研究（利用资源丰富领域的分词词性标注联合模型来帮助资源稀少领域分词词性标注），其次我们从任务上面也进行了延伸，从单独的词性标注任务到分词词性标注的联合任务，这一点是由中文的特性所引起的。对于字母型语言，没有分词这个概念存在，而扩展到中文这种非字母型语言时，就变成了分词词性标注联合模型的场景。本节中，我们首先介绍我们所使用的基本模型，然后介绍词典标注的情况及如何使用词典标注，接着介绍句子标注的情况，最后介绍我们如何进一步使用自学习（Self-Training）来提升词典标注或句子标注的性能。

### 5.4.1 基准模型

我们所使用的基本模型是张岳和Clark等人2010年提出来的基于转移柱搜索的分词词性标注联合模型[3]。在基于转移的系统中，其算法核心部分是解码过程中所处的一系列状态以及建立在每个状态上的一系列操作。在分词词性标注


图 5-1 基于转移的分词词性标注联合模型状态定义。

Fig. 5-1 State definition of the transition-based joint segmentation and POS tagging.

联合模型中，状态由一个栈和一个队列组成，栈中存储这一个已经部分解码的中文词和词性标签序列，而队列中存储的是尚未进行处理的字序列，如图5-1所示。相应每个状态的转移操作共有两类，一类是分开（Separate），另一类是附加（Append）。分开是指将队列中的第一个字$c_0$移入栈中，作为一个独立的词的开始，因此栈的长度增1；需要注意的是，词性会作为这个操作的参数赋给以$c_0$为开始汉字的词，由于每一个词都会而且仅会面临一次分开操作，因此我们最终得到的结果中的每一个词都会被唯一的赋予一个词性标记。附加是指将队列中的第一个字$c_0$移入栈中，作为栈顶中文词的最后一个字，因此栈的长度保持不变。

在具体解码时，初始状态是栈为空，队列中存储着整个句子的字序列，最终状态是队列为空，则栈中存储的词与词性的序列便是最终分词和词性标注的结果。中间解码时，每个状态都可能面临上述两个转移操作，从而生成两个新的状态。由于每个状态都可能变为两个状态，因此在到达第i步，也就是在处理到第i个字符时，所有可能的生成状态个数为$2^i$，这样在搜索和排序算法的时间和空间耗费是以指数级别增长的。为了限定算法的复杂度，我们采用了柱搜索算法，即我们每次只保留分数最高的固定大小的状态数目。联合模型中所使用的特征与张岳和Clark等人2010年的论文中所提到的特征完全一样，如表5-1所示，其中$c$，$w$和$t$分别代表字，词和词性；下标中的数字表示距离此处分析所在位置的距离；$s(w)$，$e(w)$以及$l(w)$ 表示某个词的第一个字，最后一个字和该词的长度。在训练模型参数时，联合模型采用了平均感知器算法结合提前更新。

### 5.4.2 词典标注

词典标注是采用人工标注的方法来构建一个特定规模的词典，在该词典中，每一项为一个词以及该词所有可能的词性。不少研究者也将词典标注称为类型标注（Type-Annotation）[120, 122–124]。一般情况下，词典标注的场景是给定一个词，标注者通过自己的联想为它标注所有可能的词性，这个词性的数目往往不操过三个，也是比较容易想到的词性。我们这一章的目的是为了使用词典标注来提升某一领域的性能，因此被标注的词典应该是一个领域词典，从而每

表 5-1 基于转移的分词和词性标注联合模型特征模板。

Table 5-1 Feature templates for transition-based joint word segmentation and POS tagging.

| 使用特征的动作 | 特征模板 |
|---|---|
| 分开 | $t_{-1}t_0$, $t_{-2}t_{-1}t_0$, $w_{-1}t_0$, $c_0t_0$, $s(w_{-1})\circ t_0$, $c_{-1}c_0t_{-1}t_0$, $w_{-2}w_{-1}$, $e(w_{-1})\circ t_{-1}$, $e(w_{-1})\circ c_0$, $w_{-1}c_0$, $w_{-1}t_{-1}\circ e(w_{-2})$, $w_{-1}$, $s(w_{-1})\circ l(w_{-1})$, $e(w_{-1})\circ l(w_{-1})$, $s(w_{-1})\circ e(w_{-1})$, $e(w_{-2})\circ e(w_{-1})$, $e(w_{-2})\circ w_{-1}$, $s(w_{-1})\circ c_0$, $w_{-1}\circ l(w_{-2})$, $w_{-2}\circ l(w_{-1})$, $w_{-1}t_{-2}$, $w_{-1}t_{-1}$, $w_{-1}t_{-1}c_0$, $w_{-1}$, where $l(w_{-1})=1$, $c_{-2}c_{-1}c_0t_{-1}$, where $l(w_{-1})=1$, $ct_{-1}\circ e(w_{-1})$, where $c\in w_{-1}$ , |
| 附加 | $c_0t_{-1}$, $c_{-1}c_0$, $c_{-1}c_0t_{-1}$, $s(w_{-1})\circ c_0t_{-1}$ |

一个待标注的词必须属于一个特定的目标领域，并且为该词所标注的词性也是有针对性的，必须面向该目标领域。

在我们的场景中，面临的任务是分词词性标注的联合任务，因此词也是无法预先知道的，所以上面的这个场景并不适合，因为它假定了待标注的词必须预先指定。实际操作中，我们的标注还是按句进行，但是并不标注一个句子中所有词的词性，只是有针对性的从中选择一个或者多个领域相关的词为它标注词性，在标注词性时，标注者并不一定要局限于这个词在这一句子中的词性，而是如果联想到某个词性，便就向词典中增加一个词-词性项。为了尽量减少标注所消耗的劳动力，我们使用了一个利用源领域训练语料得到的一个分词词性标注器，来对目标领域的大量未标注领域句子做一个自动分析，标注者可以在这样一个初步的结果上进行标注。

### 5.4.3 基于词典的分词词性标注联合模型

前面介绍了词典标注的工作是如何进行的，下面我们将介绍如何使用这些词典。词典的合理应用是解决分词词性联合模型领域自适应的关键。对于分词，非常自然的一种想法是将统计模型解码之后的结果根据标注词典按照一定的匹配规则进行后处理，但是这个处理过程无法考虑应用规则时所处在的上下文，因此这个后处理还会面临很多歧义。为了使得词典中的信息能够和分词过程中复杂的上下文结合在一起，我们将这种基于规则的处理和基于统计模型的处理颠倒过来，即先对当前要处理的字符片段使用基于词典的规则，获取目前的所有可能切分，然后基于这些切分，产生一系列特征，最后把这些特征融入

到统计模型中去。这样，这些基于词典的规则便能和上下文相结合，以特征的方式体现出来，而这些特征的权重，我们可以非常方便的通过统计学习得到。这些特征都是非词汇化的特征，即与词典中某个具体的词无关，这样，应用在其他领域时，我们只需将词典替换即可。

这一方法可以非常方便的扩展到分词词性标注的联合模型上面，为了更清楚的解释上面提到的方法，首先我们将中文的词分成两部分，领域相关的词和领域无关的词，然后为领域相关的词定义一类非词汇化的特征，也就是特征中不含有具体的某个词。我们利用源领域的标注句子集（Source Corpus）、源领域相关的词典（Source Lexicon）和领域无关的词典（Common Lexicon）来训练这类非词汇化的特征以及其他基本联合模型特征。在对目标领域中指定的句子进行解码时，我们使用目标领域相关的词典（Target Lexicon）和领域无关的词典（Common Lexicon），来提取非词汇化的特征。由于变换了词典，这时相关的非词汇化特征便起到了作用，使得解码时领域相关词典的信息被融入到了这个过程中。图5-2给出了我们这种方法的框架图。在训练时，我们采用源领域的语料，源领域的词典，而在实际应用时，我们使用目标领域词典进行解码。

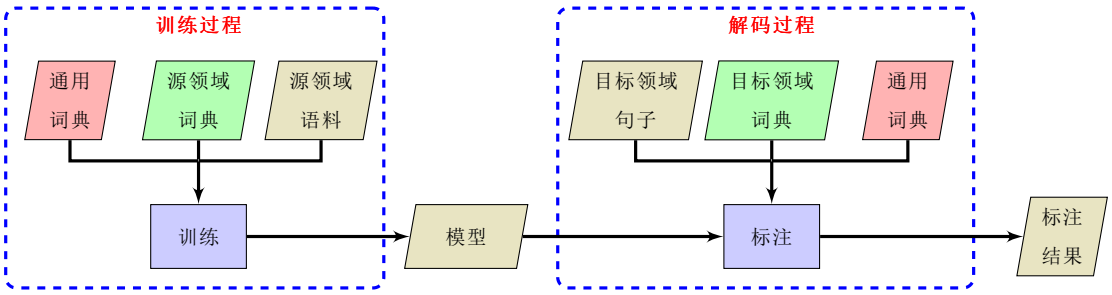



图 5-2 基于领域词典的分词词性标注联合模型的框架示意图。

Fig. 5-2 Architecture of our lexicon-based model for domain adaptation.

具体说来，我们用一个词性标注例子解释上面这个的框架，当然这个例子也很容易扩展到分词词性标注的联合模型中去。假设源领域的一个句子“江泽民|NR 随后|AD 访问|VV 上汽|NR”和目标领域的句子“碧瑶|NR 随后|AD 来到|VV 大竹峰|NR”。“江泽民”和“碧瑶”都是两个领域中的人名，同时“上汽”和“大竹峰”都是地名，如果这四个词都被认定为领域相关词时，我们将这两个句子都进行非词汇化，则这两个句子会变化为“⟨domain-NR⟩ AD VV ⟨domain-NR⟩”，从而它们具有了一个相同的形式，因此，源领域中的这个句子“江泽民|NR 随后|AD 访问|VV 上汽|NR”便能正确的分析目标领域的句子“碧瑶 随后 来到 大竹峰”。当一个句子中领域无关的词足够多时，而且领域相关的

词都存在与领域词典中时，我们的这个方法会非常有效，这个方法可以看作是把一个句子中所有领域相关词都被替换成了匹配标记，表示着它在目标领域词典中，而且它的词性是词典中给出的词性列表中的一个。

在我们使用的基于词典标注的模型中，所使用的非词汇化特征如表5-2所示，其中$w_{-1}$和$t_{-1}$表示栈顶的最后第一个词和最后第二个词，$l(w)$表示词$w$的长度；in-lex$(w,t)$表示词-词性对$(w,t)$是否在领域相关词典中。还有一个问题是如何区分领域相关的词和领域无关的词，我们采用一个非常简单的方法来区分这些词，即通过中文的知网来进行区别[44]，在知网中出现的词被认定为领域无关的词，否则便是领域相关的词。

### 5.4.4 句子标注

句子标注，也就是以句子为单位去标注目标领域中的句子，标注句子要求把句子中的每个词都识别出来，而且给出正确的词性，不像词典标注只标注句子中的部分领域相关词。因为实际标注中词性变化多的词往往是通用领域词，而且词也非常难标，因此句子标注要比词典标注难很多，而且如果标注和源领域语料存在着比较大的冲突时，会使得通用领域词的分词和词性标注性能下降。和词典标注的场景类似，我们也使用了一个利用源领域训练语料得到的一个分词词性标注器来对待标注的句子进行预处理，这样会大大的减少标注者的标注代价。标注一定量的目标领域句子，然后将这些句子和源领域的句子混在一起训练，也是一种非常有效的领域迁移方法。虽然有不少研究者提出了其它的混合训练方法，但是基本上目标领域性能提升也只是稍微超过了这一简单的混合方法[121]。

### 5.4.5 自学习算法

为了进一步加强基于词典标注和句子标注在领域自适应中的效果，我们使用了自学习的方法。通过这样的加强，我们也可以更为全面的比较词典标注和

表 5-2 基于词典标注的分词词性联合模型中所使用的非词汇化词典特征。
Table 5-2 Unlexicalized dictionary features for joint segmentation and POS tagging.

| Action | Lexicon Feature templates |
|---|---|
| ***Separate*** | in-lex$(w_{-1})$, $l(w_{-1}) \circ$ in-lex$(w_{-1})$, in-lex$(w_{-1},t_{-1})$, $l(w_{-1}) \circ$ in-lex$(w_{-1},t_{-1})$ |

句子标注的领域自适应能力。自学习算法利用目前最好的模型先自动分析一部分未标注数据，然后从中选择置信度比较高的句子，最后将这些自动分析句子加入同一个模型下一步的训练语料中。我们沿用Yang Liu和张岳提出的基于混乱度的排序算法[118]，来对自动分析的句子的置信度进行排序。假设对句子$c_1 \cdots c_n$自动解析之后的结果$w_1_t_1 \ \cdots \ w_m_t_m$，则该句子我们用公式(5-1)对其打分，分数越高，越容易被选择。

$$\text{Score}(i+1) = \text{Score}(i) + \ln P(c_{i+1}|c_{i-1}c_i) \tag{5-1}$$

其中Score(·)表示分析到指定位置时的分数，Score($n$)即为整个句子的分数，$P(c_{i+1}|c_{i-1}c_i)$由在目标领域未标注预料中根据基于字的语言模型计算获得。

将自学习算法和词典标注相结合时，首先根据目前的源领域词典$\Omega_s$和源领域训练语料$C_s$训练出一个模型$M_1$，然后将$M_1$结合目标领域词典$\Omega_t$去自动解码目标领域的未标注句子，使用上面介绍的混乱度排序算法得到置信度最高的$K$个句子，然后这$K$个句子和源领域训练语料$C_s$结合去训练被自学习增强的最终模型$M_2$，最后使用$M_2$来进行最终的解码。在训练$M_2$时，需要注意的是非词汇化特征的提取，在提取源领域训练语料$C_s$的非词汇化特征时，我们使用的是源领域词典$\Omega_s$，而提取新加入的$K$个目标领域句子的非词汇化特征时，我们使用的是目标领域词典$\Omega_t$。

将自学习算法和句子标注相结合时，简单的将源领域训练语料$C_s$和目标领域训练语料$C_t$相结合得到第一个模型$M_1$，使用$M_1$去自动解码目标领域的未标注句子，类似同样使用上面介绍的混乱度排序算法得到置信度最高的$K$个句子，训练经过自学习增强的模型$M_2$时，简单的将这两种句子合并即可，因为句子标注并不需要提取非词汇化的特征。

## 5.5 实验结果与分析

### 5.5.1 实验设置

在实验中，我们的源领域数据是中文宾州树库5.0（Chinese Penn Treebank 5.0, CTB5.0）中的数据，这些数据基本上都是属于新闻领域的；而我们的目标领域是网络小说领域，我们一共标注了4,555 句《诛仙》小说（Zhuxian，ZX）中的数据，其中部分用于开发，部分用于测试，另外部分用于句子标注的实验，表5-3 中给出了详细的数据统计信息。在评价时，我们使用基于词的准确率、召回率和它们的F值来评价分词和词性标注的性能；我们主要以评价词性

标注的性能为主，因为它代表了系统的分词词性联合准确率。

表 5-3 语料统计信息。

Table 5-3 Corpus statistics.

| 数据集 | | 章节号划分 | 句子数目 | 词数目 |
|---|---|---|---|---|
| CTB5 | 训练集 | 1-270,400-931, 1001-1151 | 10,086 | 493,930 |
| | 开发集 | 301-325 | 350 | 6,821 |
| | 测试集 | 271-300 | 348 | 8,008 |
| 诛仙 | 训练集 | 6.6-6.10,7.6-7.10 | 2,373 | 67,692 |
| | 开发集 | 6.1-6.5 | 788 | 20,3939 |
| | 测试集 | 7.1-7.5 | 1,394 | 34,355 |

### 5.5.2 数据标注介绍

这一节我们简单介绍有关数据标注的一些信息，首先我们简要介绍一下所选取的《诛仙》小说语料的特点。一方面《诛仙》小说语料属于网络小说体裁，这一小说具有一个特定的写作风格，其写作方式类似与明清小说的古文体；另一方面《诛仙》小说语料的题材是一个玄幻故事。我们在表5-4中给出了中文宾州树库的一些典型句子以及《诛仙》小说中的一些典型句子，通过这些句子的比较，我们可以直观的体会这两种语料的区别。通过这个比较我们也可以发现领域迁移问题是一个非常复杂的问题，领域这个词反应在具体情况时，可以是新闻、金融、生物、化学等，这属于题材的不同，还有它可以反应为网络体，记叙文、议论文、小说等等的不同，属于体裁的不同。在《诛仙》语料中，我们能观察到这两个方面的不同。

除了领域之间的差异，我们在这里需要重点介绍的是关于词典标注和句子标注的时间差异。在标注句子时，我们一共标注了4555个句子，大约花了80小时，平均一分钟一个句子。为了进行基于词典的领域自适应，我们花了五个小时标注了3,000个词-词性配对，使用同样的时间标注者能大概标注300个句子。

### 5.5.3 基本模型性能

我们所使用的基本模型是目前最好的系统之一，它在CTB5上取得了97.62%分词准确率以及93.85%的分词词性联合准确率，当这个模型应用

表 5-4 CTB5语料和诛仙语料中的反应两者差别的典型例句。

Table 5-4 Example sentences from CTB5 and ZX to illustrate the differences between them.

| **CTB5中的句子** | **诛仙小说中的句子** |
|---|---|
| 乔石会见俄罗斯议员团 | 天下之大，无奇不有，山川灵秀，亦多妖魔鬼怪。 |
| 李鹏强调要加快推行公务员制度 | 时间无多，我去请出诛仙古剑。 |
| 法正研究从波黑撤军计划 | 张小凡心头恍惚，如梦似幻! |
| 第七届世界游泳锦标赛在罗马开幕 | 夜色深沉，苍穹无语! |
| 中国化学工业加快对外开放步伐 | 忽听得狂笑风起，法宝异光闪动。 |

在诛仙语料上时，系统的性能下降了不少，仅取得了87.71%的分词准确率和80.81%的分词词性联合准确率，最终的词性标注性能下降大约为13%。我们进一步使用了自学习的方法，这一方法与Yang Liu和张岳2012年文中的方法一样[118]，最终我们的基本模型被提升到了88.62%分词准确率以及81.94%的分词词性联合准确率，在词性标注上的提升幅度大概为1%，这也说明了单纯的半指导的学习方法能带来的性能提升是非常有限的。

### 5.5.4 开发集上的实验

这一节中，我们通过实验对基于词典标注的领域自适应和基于句子标注的领域自适应进行分析和比较，主要分为五个方面。首先我们观察词典标注对领域自适应的影响；进一步我们观察句子标注对领域自适应的影响；然后比较在同样的标注时间下，句子标注和词典标注谁能取得更好的性能；紧接着我们研究在相同的时间条件下，词典标注和句子标注相结合能否取得更好的效果；最后我们再利用自学习算法，将领域自适应的性能提升到最佳，并且观察自学习算法在词典标注和句子标注上的进一步提升效果，并确定我们最终的领域自适应方案。

#### 5.5.4.1 词典对基于领域自适应的影响

第一步，我们验证基于词典的方法是否会给分词词性标注的领域迁移问题带来一定效果，并且调查词典的大小对领域自适应的影响。在实验中，我们用了三种词典，第一种是从百度百科对该小说的相关介绍中直接提取的人名、地点名、武器名等专有名词词典，这类专有名词的词性被直接赋予NR，我们用记号***NR***来表示这类词典；第二种词典是标注的3000个词和词性的配对，加入***NR***词典中，将这个词典命名为***3K***，第三种词典是直接从开发数据集中将所

有词以及词性提取出来，组成一个词典，相当于这种基于词典的模型能达到的最好性能，这个词典我们命名为***ORCALE***，我们将这三个词典的性能进行对比，验证是否随着词典规模的增加，目标领域的性能会逐步增加。

表5-5中“词典标注”和“原始模型”的交叉部分给出了基于词典标注的模型在开发集上的最终结果，从表中的结果可以看出，词典标注确实能增加目标领域的性能，而且随着词典规模的扩大，或者词典对开发集数据覆盖得越全面，目标领域性能也越高。我们可以看到，随着词典的变大，或者说，词典对测试集的覆盖越全面，目标领域的性能也越高。特别的是，我们发现经过一个人五个小时的标注代价，这种基于词典标注的领域自适应方法，能将《诛仙》领域开发集词性标注从从来的82.92%提升到86.53%，而且我们还发现即便使用开发集中所有的词和词性构成的领域词典也只能将其性能提升到88.87%，这反映了单纯词典标注的一个极限。

#### 5.5.4.2 句子标注对领域自适应的影响

我们利用目标领域标注语料的训练语料部分，来调研句子标注对领域自适应的影响。最主要的目的是为了了解标注规模的大小对目标领域性能的提升程度，也就是标注的句子数目对目标领域性能的影响。我们分别使用300句、600句以及900句《诛仙》训练语料，并将它们和源领域训练语料结合在一起训练（这样能取得更好的结果），来观察目标领域的性能。最终的结果如表5-5所示（句子标注和原始模型的交叉部分），通过表中的数据，我们可以发现，从基本模型到300句，从300句到600句以及600句到900句，每逐步增加300句，虽然目标领域的性能是在逐步提升，但是提升的幅度逐步减小，从600句到900句仅有0.4%的词性准确率提升。

#### 5.5.4.3 词典标注和句子标注相比较

从表5-5中的代价数据显示，标注300个句子和标注词典***3K***所需要的代价几乎是相同的。另外从前面的实验结果上我们可以发现，标注300个句子所得到的准确率与注词典***3K***所得到的准确率也是比较接近的，300个句子在开发集上能取得92.59%的分词准确率和86.86%的词性标注准确率，而词典***3K***能取得91.93%的分词准确率和86.53%的词性标注准确率，略低于300个句子所带来的性能。因此我们可以认为，词典标注和句子标注在基于同样的代价下它们能等价的增强分词词性标注联合模型的领域自适应能力。

实际上，由于网络小说《诛仙》出自一人之手，其写作风格相对固定，因此在考虑增强联合模型的领域自适应能力时，语言风格问题是一个首要的因素，而句子标注最能体现语言风格问题，因为我们能从句子标注中学得比较完

表 5-5 开发集上的测试结果。

Table 5-5 Development test results.

| 模型 | 目标领域资源 | 代价 | 原始模型 | | +自学习 | | |
|---|---|---|---|---|---|---|---|
| | | | 分词 | 词性 | 分词 | 词性 | ER |
| 基本模型 | — | 0 | 89.77 | 82.92 | 90.35 | 83.95 | 6.03 |
| 词典标注 | NR(T) | 0 | 89.84 | 83.91 | 91.18 | 85.22 | 8.14 |
| | 3K(T) | 5h | 91.93 | 86.53 | 92.86 | 87.67 | 8.46 |
| | ORACLE(T) | ∞ | 93.10 | 88.87 | 94.00 | 89.91 | 9.34 |
| 句子标注 | 300(S) | 5h | 92.59 | 86.86 | 93.33 | 87.85 | 7.53 |
| | 600(S) | 10h | 93.19 | 88.13 | 93.81 | 89.01 | 7.41 |
| | 900(S) | 15h | 93.53 | 88.53 | 94.15 | 89.33 | 6.97 |
| 词典和句子标注相结合 | 3K(T) + 300(S) | 10h | 93.49 | 88.54 | 94.00 | 89.21 | 5.85 |
| | 3K(T) + 600(S) | 15h | 93.98 | 89.27 | 94.61 | 89.87 | 5.59 |

整的写作风格，因此它比较占优势。虽然我们很难从词典标注中学习出这种语言风格来，但是我们也发现词典标注和句子标注对目标领域的性能提升是几乎等价的，这一结果也和Dan Garrette 等人的论文中的结论一致，即词典标注往往会比句子标注更有效[122]。

词典标注的特点能更广的覆盖目标领域句子中的未登录词，而句子标注的特点能更好的捕捉目标领域句子语言模型，因此这两者标注应该是互补的。更具体的，我们通过散点图对比了一下等时间的基于词典的标注和基于句子的标注两者在开发集上的错误分布，如图5-3所示，其中横坐标是使用300个目标领域句子训练出来的模型在某个句子上的词性性能，纵坐标是同样的句子在使用目标领域词典**3K**的词性性能，两种标注都需要5个小时的标注时间。从图中我们可以看到，散点图上点分布在对角线两侧非常均匀，比较少有点集中在对角线上，这表明两者错误分布完全不一样，如果将两者进行结合，领域自适应的效果将会更进一步增加。

#### 5.5.4.4 句子标注和词典标注相结合

前面的比较我们已经提到了，词典标注和句子标注应该是互补的，两者结合能取得更好的效果，因此这一节的实验我们将讨论这样一个问题：词典标注和句子标注相结合，能否取得比任何单一标注更好的性能?

我们分别将目标领域词典***3K***和目标领域300个句子以及600个句子相结合，

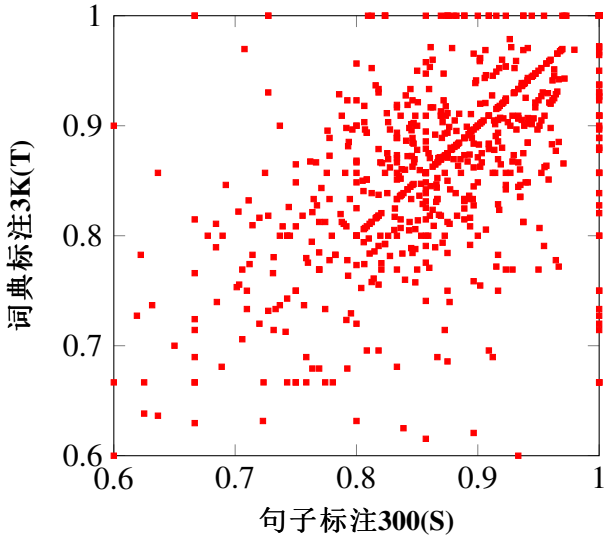



图 5-3 使用300个目标领域句子和词典***3K***时，开发集上性能对比散点图。

Fig. 5-3 Scatter plots between model with 300 sentences and model with ***3K*** lexicon.

其消耗的时间分别为10小时和15小时，在这种情况下我们分别和只标注600个句子和900个句子的性能相比较（这两种情况所消耗的标注时间也是10小时和15小时，而且由于词典标注存在着一个最高性能，这个最高性能仅和900个句子标注接近，因此随着句子标注的增加，结合的方法肯定由于单一的句子标注），其最终结果如表5-5 最下面的部分所示。目标领域词典***3K***和目标领域300个句子相结合，能取得93.49%的分词准确率和88.54%的词性标注准确率，而同标注代价的600个句子标注，能取得93.19%的分词准确率和88.13%的词性标注准确率，因此两者结合的方法能稍稍优于单一的句子标注。目标领域词典***3K***和目标领域600个句子相结合，能取得93.98%的分词准确率和89.27%的词性标注准确率，而同标注代价的900个句子标注，能取得93.53%的分词准确率和88.53%的词性标注准确率，同样两者结合的方法能优于单一的句子标注。另外我们也可以发现，当标注句子数目增加到一定程度的时候，词典标注能取得更明显的效果。

#### 5.5.4.5 结合自学习

我们进一步使用自学习来提升基于词典标注和基于句子标注的领域自适应的性能。表5-5 中显示了自学习带来的提升效果，ER表示自学习带来的错误减少百分比。首先，我们使用的基本模型也能在自学习的帮助下取得6.03%的错误减少，这个错误减少数要比基于句子标注的错误减少数少一点，基于句子标注的模型在自学习的帮助下能带来平均7.3%的错误减少，这个数目又小于自学

习为基于词典标注带来的错误减少，平均为8.7%左右。

自学习所带来的好处主要是能更准确的估计目标领域的语言模型，因为自学习得到的是自动标注句子，因此它能取得和句子标注类似的效果，从而导致自学习在基于句子标注上取得的效果比不上在基于词典标注上取得的效果。但是基准模型由于获取领域语言模型相关的知识非常有限，因此带来的提升也不高。自学习也能为句子标注和词典标注的结合模型带来一定效果，但是由于结合的标注方法本身性能已经很高了，所以性能的提升效果也不那么明显。

### 5.5.5 最终测试结果

表 5-6 测试集上的最终结果。

Table 5-6 Final results on test data set.

| | 分词 | 词性 | ER | 代价 |
|---|---|---|---|---|
| 基准模型 | 87.71 | 80.81 | 0.00 | 0 |
| 基准模型+自学习 | 88.62 | 81.94 | 5.89 | 0 |
| **词典标注** | | | | |
| NR(T) | 88.34 | 82.54 | 9.02 | 0 |
| NR(T)+自学习 | 89.52 | 83.93 | 16.26 | 0 |
| 3K(T) | 91.11 | 86.04 | 27.25 | 5h |
| 3K(T)+自学习 | 92.11 | 87.14 | 32.99 | 5h |
| **句子标注** | | | | |
| 300(S) | 92.44 | 86.87 | 31.58 | 5h |
| 300(S)+自学习 | 93.24 | 87.48 | 34.76 | 5h |
| 600(S) | 93.09 | 88.05 | 37.73 | 10h |
| 600(S)+自学习 | 93.77 | 88.78 | 41.53 | 10h |
| **词典和句子标注相结合** | | | | |
| 3K(T)+300(S) | 93.27 | 89.03 | 42.83 | 10h |
| 3K(T)+300(S)+自学习 | **93.98** | **89.84** | **47.06** | 10h |

表5-6显示了我们的基于词典标注、句子标注以及两者相结合的方法在最终测试集上的性能，其中代价是指所用到的资源所需要的标注时间。表中的数据表明，通过一个人一天10个小时的标注，结合自学习方法，诛仙目标领域

的性能从最初的分词准确率87.71%、词性准确率80.81%提升到最终的分词准确率93.98%、词性准确率89.84%，分词提升了6.27%，词性提升了9.03%。表中的结果也表明了词典标注和句子标注相结合能带来最好的目标领域性能。

## 5.6 本章小结

本章我们为分词词性标注的联合模型提出了一种领域自适应的方法，基本思想通过标注少量的目标领域数据，使得目标领域的性能有了很大的提升。对目标领域的数据标注不是简单标注目标领域的句子，而是通过词典标注和句子标注相结合。

词典标注和句子标注都是提升统计模型领域自适应能力的非常有效的手段，前人的工作大部分关注于单独的比较这两种方法，而我们进一步深入考虑，将这两种标注方法结合。我们的实验结果也表明了在相同的标注代价下，结合的方法能取得更好的性能。

我们的方法也可以非常方便的扩展到其它联合模型上面，联合模型和普通模型相比，由于它要同时标注两种以上的数据信息，因此标注的代价会更高，句子标注的代价是非常昂贵的，在分词词性标注中的词典标注相当于对句子的部分结构标注，或者是小结构标注，然后我们收集所有的部分小结构标注，构成一个词典，然后把这种词典通过非词汇化或匹配的方式融入到统计模型中，便形成了基于词典标注的联合模型。因此对于其它的联合模型，我们也可以使用同样的方法，将句子全部标注和句子的部分结构标注即词典标注相结合。

# 结 论

词法分析、句法分析和语义分析是中文句子级别分析的三类基本分析技术，其中词法分析主要包括分词和词性标注，句法分析可以是短语结构句法分析也可以是依存结构句法分析，语义分析在本文中主要是指语义依存分析。这些基本分析技术能为自然语言处理相关应用（例如问答，机器翻译和信息检索等等）提供基本的输入信息，因此提升这些句子级别的基本分析技术的性能有着很重要的意义。

联合模型的方法通过将多个层级相邻的任务结合在一起采用同一个模型进行同时处理，能有效的避免传统串行模型所存在的错误蔓延和逐层局部优化两个方面的问题。另外联合模型还能为自然语言处理的研究者提供了一个非常方便的理解词法、句法以及语义之间相互关系的手段。对于联合模型的研究，我们首先需要考虑的问题是如何建模，也就是建模方法的问题。其次，考虑到多个任务的联合处理往往会带来更高的解码复杂度，从而导致了联合模型在效率上的不足，因此联合模型性能和效率之间的平衡也是一个非常重要的问题。另外，联合模型和其他自然语言的模型一样，都存在着一个非常难而且普遍的领域自适应问题，我们在本论文也对这一问题进行了研究。这三个方面的问题是对联合模型研究层面逐步提高的一个过程。本文的主要研究内容和研究成果覆盖了这三个方面的问题，具体包括如下四个方面：

首先，针对词法分析和句法分析联合模型的建模方法问题，本文利用大部分中文词语存在着内部结构这一特点，扩展了传统基于词的句法分析方法，提出了基于字的中文句法分析方法，从而非常自然的将词法分析和句法分析联合在一起，得到了中文词法句法的联合模型。实验结果表明，这种基于字的分析方法能有效的提升中文词法句法的性能，取得了目前最好的结果。

其次，针对句法分析和语义分析联合模型的建模方法问题，本文采用语义依存分析这一语义分析手段，使得语义分析和句法分析的联合变得非常自然。我们首先从理论和实验两方面表明了语义依存分析作为语义分析手段的合理性，然后在此基础上我们提出了语义依存分析和句法依存分析的联合模型，并且通过实验表明了这一联合模型的优越性。

再次，针对词性标注和依存句法联合模型的低效率问题，我们使用了一种模型融合和过训练相结合的方法，使得这一联合模型的速度增加了十倍以上但是性能仍然能几乎不变。这一方法具有很好的通用性，一方面模型融合使得联

合模型性能进一步提升但是效率进一步下降，而另一方面过训练使得一个高效率低性能的联合模型在前面融合模型的帮助下得到了大幅度的提升。

最后，针对于分词词性标注联合模型的领域迁移问题，我们采用了语料标注的方法。语料标注的手段一般分为两种，以句子为单位标注和以一些公用片段为单位标注；在分词词性标注的联合中，前者反映为句子标注而后者反映为词典标注。在本文的方法中，我们将这两种标注相结合，实验结果表明这种结合的方法在指定标注代价下能取得更有效的领域自适应效果。

需要说明的是，前面的两个研究内容是针对联合模型的建模而提出的；第三个研究内容是在考虑联合模型性能和效率之间的一个平衡，我们以词性标注和依存句法的联合模型为例来研究了这一问题，实际上我们提出的方法具有很好的通用性，可以非常容易的扩展到其他联合模型中；第四个研究内容是针对联合模型的领域自适应而开展的，考虑到这一问题具有非常大的难度，因此我们以最简单的分词词性标注联合模型为例展开了这方面的工作，我们提出的方法也具有通用性，但是扩展到其它任务上。

综上所述，本文在中文句子级别分析技术的联合模型上做了一些尝试并取得了一些初步的成果，然而，联合模型在自然语言处理领域仍然是一个非常活跃而且具有很大挑战的课题，目前的研究还远远不够。因此需要更多的研究工作继续进行深入研究，尤其是面对其它自然语言处理的任务联合时，还有很多问题需要解决。

# 参考文献

[1] Jiang W, Huang L, Liu Q, et al. A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging[C]. Proceedings of ACL-08: HLT. 2008:897–904.

[2] Zhang Y, Clark S. Joint Word Segmentation and POS Tagging Using a Single Perceptron[C]. Proceedings of ACL-08: HLT. 2008:888–896.

[3] Zhang Y, Clark S. A Fast Decoder for Joint Word Segmentation and POS-Tagging Using a Single Discriminative Model[C]. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. 2010:843–852.

[4] Li Z, Zhang M, Che W, et al. Joint Models for Chinese POS Tagging and Dependency Parsing[C]. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011:1180–1191.

[5] Berger A L, Pietra V J D, Pietra S A D. A maximum entropy approach to natural language processing[J]. Computational Linguistics, 1996, 22(1):39–71.

[6] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3):1–27.

[7] Collins M. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms.[C]. Proceedings of the 7th EMNLP. 2002.

[8] Collins M, Roark B. Incremental Parsing with the Perceptron Algorithm[C]. Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume. 2004:111–118.

[9] Xue N. Chinese word segmentation as character tagging[J]. International Journal of Computational Linguistics and Chinese Language Processing, 2003, 8(1).

[10] Tseng H, Chang P, Andrew G, et al. A conditional random field word segmenter for sighan bakeoff 2005[C]. Proceedings of the fourth SIGHAN workshop. 2005:168–171.

[11] Low J K, Ng H T, Guo W. A Maximum Entropy Approach to Chinese Word Segmentation[C]. Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing. 2005:161–164.

[12] Gao J, Li M, Wu A, et al. Chinese word segmentation and named entity recognition: A pragmatic approach[J]. Computational Linguistics, 2005, 31(4):531–574.

[13] Zhang Y, Clark S. Chinese Segmentation with a Word-Based Perceptron Algorithm[C]. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. 2007:840–847.

[14] Zhang R, Kikui G, Sumita E. Subword-Based Tagging for Confidence-Dependent Chinese Word Segmentation[C]. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. 2006:961–968.

[15] Sun W. Word-based and Character-based Word Segmentation Models: Comparison and Combination[C]. Coling 2010: Posters. 2010:1211–1219.

[16] Banko M, Moore R C. Part-of-Speech Tagging in Context[C]. COLING. 2004.

[17] Huang Z, Eidelman V, Harper M. Improving A Simple Bigram HMM Part-of-Speech Tagger by Latent Annotation and Self-Training[C]. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. 2009:213–216.

[18] Owoputi O, O'Connor B, Dyer C, et al. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters[C]. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013:380–390.

[19] Sutton C, McCallum A. An introduction to conditional random fields[J]. Machine Learning, 2011, 4(4):267–373.

[20] Søgaard A. Part-of-speech tagging with antagonistic adversaries[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2013:640–644.

[21] Tsuruoka Y, Miyao Y, Kazama J. Learning with Lookahead: Can History-Based Models Rival Globally Optimized Models?[C]. Proceedings of the Fifteenth Conference on Computational Natural Language Learning. 2011:238–246.

[22] Collins M. Head-Driven Statistical Models for Natural Language Parsing[D]. Pennsylvania University, 1999.

[23] Charniak E, Johnson M. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking[C]. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). 2005:173–180.

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

[25] Zhang Y, Clark S. Syntactic Processing Using the Generalized Perceptron and Beam Search[J]. Computational Linguistics, 2011, 37(1):105–151.

[26] Zhu M, Zhang Y, Chen W, et al. Fast and Accurate Shift-Reduce Constituent Parsing[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013:434–443.

[27] Klein D, Manning C D. Accurate Unlexicalized Parsing[C]. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. 2003:423–430.

[28] Matsuzaki T, Miyao Y, Tsujii J. Probabilistic CFG with Latent Annotations[C]. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). 2005:75–82.

[29] Petrov S, Barrett L, Thibaux R, et al. Learning Accurate, Compact, and Interpretable Tree Annotation[C]. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. 2006:433–440.

[30] Petrov S, Klein D. Improved Inference for Unlexicalized Parsing[C]. Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference. 2007:404–411.

[31] McDonald R, Crammer K, Pereira F. Online large-margin training of dependency parsers[C]. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05. 2005:91–98.

[32] Carreras X. Experiments with a higher-order projective dependency parser[C]. Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL. 2007:957–961.

[33] Koo T, Collins M. Efficient third-order dependency parsers[C]. Proceedings of the 48th Annual Meeting of the ACL. 2010:1–11.

[34] Zhang H, McDonald R. Generalized Higher-Order Dependency Parsing with Cube Pruning[C]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012:320–331.

[35] Nivre J. Algorithms for Deterministic Incremental Dependency Parsing[J]. Computational Linguistics, 2008, 34(4):513–553.

[36] Goldberg Y, Nivre J. A Dynamic Oracle for Arc-Eager Dependency Parsing[C]. Proceedings of COLING 2012. 2012:959–976.

[37] Sartorio F, Satta G, Nivre J. A Transition-Based Dependency Parser Using a Dynamic Parsing Strategy[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013:135–144.

[38] Zhang Y, Clark S. A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing[C]. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. 2008:562–571.

[39] Huang L, Jiang W, Liu Q. Bilingually-constrained (monolingual) shift-reduce parsing[C]. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. 2009:1222–1231.

[40] Zhang Y, Nivre J. Transition-based Dependency Parsing with Rich Non-local Features[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011:188–193.

[41] Sun W. Learning Chinese Language Structures with Multiple Views[D]. Saarland University, 2012.

[42] Sun W, Wan X. Data-driven, PCFG-based and Pseudo-PCFG-based Models for Chinese Dependency Parsing[J]. Transactions of the Association for Computational Linguistics (TACL), 2013, 1(1):301–314.

[43] Fellbaum C. WordNet: An Electronic Lexical Database.[M]. The MIT Press, Cambridge, Massachusetts, 1998.

[44] Dong Z, Dong Q. Hownet And the Computation of Meaning[M]. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2006.

[45] 梅家驹, 竺一鸣, 高蕴琦, et al. 同义词词林[M]. 上海辞书出版社, 1996.

[46] Tanigaki K, Shiba M, Munaka T, et al. Density Maximization in Context-Sense Metric Space for All-words WSD[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013:884–893.

[47] Bhingardive S, Shaikh S, Bhattacharyya P. Neighbors Help: Bilingual Unsupervised WSD Using Context[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2013:538–542.

[48] Carpenter B. Type-Logical Semantics[M]. The MIT Press, 1997.

[49] Zettlemoyer L, Collins M. Online Learning of Relaxed CCG Grammars for Parsing to Logical Form[C]. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007:678–687.

[50] Liang P, Jordan M, Klein D. Learning Dependency-Based Compositional Semantics[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011:590–599.

[51] Xue N, Palmer M. Annotating the Propositions in the Penn Chinese Treebank[C]. Proceedings of the Second SIGHAN Workshop on Chinese Language Processing. 2003.

[52] Xue N. Labeling chinese predicates with semantic roles[J]. Comput. Linguist., 2008, 34:225–255.

[53] Sun W, Sui Z, Wang M, et al. Chinese Semantic Role Labeling with Shallow Parsing[C]. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. 2009:1475–1483.

[54] Zhuang T, Zong C. A Minimum Error Weighting Combination Strategy for Chinese Semantic Role Labeling[C]. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010). 2010:1362–1370.

[55] Li M, Li J, Dong Z, et al. Building a large Chinese corpus annotated with semantic dependency[C]. Proceedings of the second SIGHAN workshop on Chinese language processing. 2003.

[56] Yan J. Chinese Semantic Dependency Analysis and Its Application[D]. University of Tokushima, 2007.

[57] Che W, Zhang M, Shao Y, et al. SemEval-2012 Task 5: Chinese Semantic Dependency Parsing[C]. Proceedings of SemEval 2012. 2012:378–384.

[58] Ng H T, Low J K. Chinese Part-of-Speech Tagging: One-at-a-Time or All-at-Once? Word-Based or Character-Based?[C]. Proceedings of EMNLP 2004. 2004:277–284.

[59] 张开旭. 使用压缩表示的中文分词词性标注研究[D]. 清华大学, 2012.

[60] Sun W. A Stacked Sub-Word Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011:1385–1394.

[61] Zhenghua Li W C T L, Min Zhang. A Separately Passive-Aggressive Training Algorithm for Joint POS Tagging and Dependency Parsing[C]. In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012). 2012:1681–1698.

[62] 李正华. 汉语依存句法分析关键技术研究[D]. 哈尔滨工业大学, 2013.

[63] Li Z, Zhang M, Che W, et al. Joint Optimization for Chinese POS Tagging and Dependency Parsing[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, 22(1):274–286.

[64] Hatori J, Matsuzaki T, Miyao Y, et al. Incremental Joint POS Tagging and Dependency Parsing in Chinese[C]. Proceedings of 5th International Joint Conference on Natural Language Processing. 2011:1216–1224.

[65] Bohnet B, Nivre J. A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing[C]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012:1455–1465.

[66] Luo X. A Maximum Entropy Chinese Character-Based Parser[C]. . Collins M, Steedman M. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. 2003:192–199.

[67] Li Z. Parsing the Internal Structure of Words: A New Paradigm for Chinese Word Segmentation[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011:1405–1414.

[68] 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. 2012:501–511.

[69] Hatori J, Matsuzaki T, Miyao Y, et al. Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese[C]. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2012:1045–1053.

[70] Li Z, Zhou G. Unified Dependency Parsing of Chinese Morphological and Syntactic Structures[C]. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012:1445–1454.

[71] Llus X, Carreras X, Marquez L. Joint Arc-factored Parsing of Syntactic and Semantic Dependencies[J]. Transactions of the Association for Computational Linguistics (TACL), 2013, 1(1):219–230.

[72] Emerson T. The second international Chinese word segmentation bakeoff[C]. Proceedings of the Second SIGHAN Workshop on Chinese Language Processing. 2005:123–133.

[73] Zhao H. Character-Level Dependencies in Chinese: Usefulness and Learning[C]. Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). 2009:879–887.

[74] Zhang Y, Clark S. Transition-Based Parsing of the Chinese Treebank using a Global Discriminative Model[C]. Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09). 2009:162–171.

[75] Harper M, Huang Z. Chinese Statistical Parsing[J]. Handbook of Natural Language Processing and Machine Translation, 2011.

[76] Kruengkrai C, Uchimoto K, Kazama J, et al. An Error-Driven Word-Character Hybrid Model for Joint Chinese Word Segmentation and POS Tagging[C]. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 2009:513–521.

[77] Wang Y, Kazama J, Tsuruoka Y, et al. Improving Chinese Word Segmentation and POS Tagging with Semi-supervised Methods Using Large Auto-Analyzed Data[C]. Proceedings of 5th International Joint Conference on Natural Language Processing. 2011:309–317.

[78] 赵静. 大规模汉语语义词典构建[D]. 哈尔滨工业大学, 2011.

[79] Mikolov T. Statistical language models based on neural networks[D]. Ph. D. thesis, Brno University of Technology, 2012.

[80] Huang E H, Socher R, Manning C D, et al. Improving word representations via global context and multiple word prototypes[C]. Proceedings of the 50th Annual

Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. 2012:873–882.

[81] Zelle J M, Mooney R J. Learning to Parse Database Queries using Inductive Logic Programming[C]. AAAI. 1996:1050–1055.

[82] Robert J D, Moore R, Andry F, et al. Interleaving Syntax And Semantics In An Efficient Bottom-Up Parser[C]. In Proc. of ACL-94. 1994:110–116.

[83] Wong Y W, Mooney R. Learning for Semantic Parsing with Statistical Machine Translation[C]. Proceedings of the Human Language Technology Conference of the NAACL, Main Conference. 2006:439–446.

[84] Zettlemoyer L S, Collins M. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars[C]. In Proceedings of the 21st Conference on Uncertainty in AI. 2005:658–666.

[85] Artzi Y, Zettlemoyer L. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions[J]. Transactions of the Association for Computational Linguistics, 2013, 1(1):49–62.

[86] 王丽杰. 汉语语义依存分析研究[D]. 哈尔滨工业大学, 2010.

[87] Johansson R, Nugues P. Extended Constituent-to-dependency Conversion for English[C]. Proceedings of NODALIDA 2007. 2007.

[88] de Marneffe M C, Manning C D. The Stanford Typed Dependencies Representation[C]. Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation. 2008:1–8.

[89] Chang P C, Tseng H, Jurafsky D, et al. Discriminative Reordering with Chinese Grammatical Relations Features[C]. Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation. 2009.

[90] Elming J, Johannsen A, Klerke S, et al. Down-stream effects of tree-to-dependency conversions[C]. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013:617–626.

[91] Zhou Q, Zhang L, Liu F, et al. Zhou qiaoli: A divide-and-conquer strategy for semantic dependency parsing[C]. *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. 2012:506–513.

[92] Wu Z, Wang X, Li X. Zhijun Wu: Chinese Semantic Dependency Parsing with Third-Order Features[C]. *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. 2012:430–434.

[93] Tang G, Li B, Xu S, et al. NJU-Parser: Achievements on Semantic Dependency Parsing[C]. *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. 2012:519–523.

[94] Xiong H, Liu Q. ICT:A System Combination for Chinese Semantic Dependency Parsing[C]. *SEM 2012: The First Joint Conference on Lexical and Computational Semantics. 2012:514–518.

[95] 鲁川. 现代汉语的语义网络[M]. 电子工业出版社, 1995.

[96] 袁毓林. 基于认知的汉语计算机语言学研究[M]. 北京大学出版社, 2008.

[97] 冯志伟. 中文信息处理与汉语研究[M]. 北京：商务出版社, 1992.

[98] 林杏光. 词汇语义和计算语言学[M]. 北京:语文出版社, 1999.

[99] McDonald R, Nivre J. Analyzing and Integrating Dependency Parsers[J]. Computational Linguistics, 2011, 37(1):197–230.

[100] Che W, Zhang M, Liu T, et al. A Hybrid Convolution Tree Kernel for Semantic Role Labeling[C]. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. 2006:73–80.

[101] Punyakanok V, Roth D, tau Yih W. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling[J]. Computational Linguistics, 2008, 34(2):257–287.

[102] Hajič J, Ciaramita M, Johansson R, et al. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages[C]. Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task. 2009:1–18.

[103] Sun W. Improving Chinese Semantic Role Labeling with Rich Syntactic Features[C]. Proceedings of the ACL 2010 Conference Short Papers. 2010:168–172.

[104] Laparra E, Rigau G. ImpAr: A Deterministic Algorithm for Implicit Semantic Role Labelling[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013:1180–1189.

[105] Kozhevnikov M, Titov I. Cross-lingual Transfer of Semantic Role Labeling Models[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013:1190–1200.

[106] Che W, Li Z, Li Y, et al. Multilingual Dependency-based Syntactic and Semantic Parsing[C]. Proceedings of CoNLL 2009. 2009:49–54.

[107] Wolpert D H. Stacked generalization[J]. Neural Networks, 1992, 5:241–259.

[108] Li Z, Liu T, Che W. Exploiting Multiple Treebanks for Parsing with Quasi-synchronous Grammars[C]. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2012:675–684.

[109] Nivre J, McDonald R. Integrating Graph-Based and Transition-Based Dependency Parsers[C]. Proceedings of ACL-08: HLT. 2008:950–958.

[110] Martins A F T, Das D, Smith N A, et al. Stacking Dependency Parsers[C]. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. 2008:157–166.

[111] Petrov S, Chang P C, Ringgaard M, et al. Uptraining for Accurate Deterministic Question Parsing[C]. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. 2010:705–713.

[112] McDonald R. Discriminative learning and spanning tree algorithms for dependency parsing[D]. University of Pennsylvania, 2006.

[113] Breiman L. Stacked regressions[J]. Machine Learning, 1996, 24:49–64.

[114] Charniak E. A Maximum-Entropy-Inspired Parser[C]. Proceedings of the Second Meeting of North American Chapter of Association for Computational Linguistics (NAACL2000). 2000.

[115] Che W, Wang M, Manning C D, et al. Named Entity Recognition with Bilingual Constraints[C]. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013:52–62.

[116] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2):123–140.

[117] Li Z, Che W, Liu T. Improving Chinese POS Tagging with Dependency Parsing[C]. Proceedings of 5th International Joint Conference on Natural Language Processing. 2011:1447–1451.

[118] Liu Y, Zhang Y. Unsupervised Domain Adaptation for Joint Segmentation and POS-Tagging[C]. Proceedings of COLING 2012: Posters. 2012:745–754.

[119] Sun W, Xu J. Enhancing Chinese Word Segmentation Using Unlabeled Data[C]. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011:970–979.

[120] Garrette D, Baldridge J. Type-Supervised Hidden Markov Models for Part-of-Speech Tagging with Incomplete Tag Dictionaries[C]. EMNLP-CoNLL. 2012:821–831.

[121] Daume III H. Frustratingly Easy Domain Adaptation[C]. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. 2007:256–263.

[122] Garrette D, Baldridge J. Learning a Part-of-Speech Tagger from Two Hours of Annotation[C]. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013:138–147.

[123] Garrette D, Mielens J, Baldridge J. Real-World Semi-Supervised Learning of POS-Taggers for Low-Resource Languages[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013:583–592.

[124] Täckström O, Das D, Petrov S, et al. Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging[C]. Transactions of the ACL. 2013.

# 攻读博士学位期间发表的论文及其他成果

## （一）发表的学术论文


[1] **Meishan Zhang**, Yue Zhang, Wanxiang Che, Ting Liu. *Character-Level Chinese Dependency Parsing.* In Proceedings of the 52th Annual Meeting of the Association of Computational Linguistics (ACL 2014). 2014.06, Baltimore, Maryland.

[2] **Meishan Zhang**, Yue Zhang, Wanxiang Che, Ting Liu. *Chinese Parsing Exploiting Characters.* In Proceedings of the 51th Annual Meeting of the Association of Computational Linguistics (ACL 2013). 2013.08, pp. 125-134, Sofia, Bulgaria.

[3] **Meishan Zhang**, Wanxiang Che, Yanqiu Shao, Ting Liu. *Jointly or Separately: Which is Better for Parsing Heterogeneous Dependencies?* In Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014). 2014.08, Dublin, Ireland.

[4] **Meishan Zhang**, Wanxiang Che, Ting Liu, Zhenghua Li. *Stacking Heterogeneous Joint Models of Chinese POS Tagging and Dependency Parsing.* In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012). 2012.12, pp. 3071-3088, Mumbai, India. (**EI Indexed: 20131816299391**)

[5] **Meishan Zhang**, Yue Zhang, Wanxiang Che, Ting Liu. *Type-Supervised Domain Adaptation for Joint Segmentation and POS-Tagging.* In Proceedings of the 14th Conference of the European Chapter of the ACL (EACL 2014). 2014.04, pp. 588-597，Gothenburg, Sweden.

[6] **Meishan Zhang**, Yue Zhang, Wanxiang Che, Ting Liu. *A Semantics Oriented Grammar for Chinese Treebanking*. In Proceedings of 15th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing). 2014.04, pp. 366-378, Kathmandu, Nepal. (**EI Indexed: 20142017719387**)

[7] **Meishan Zhang**, Wanxiang Che, Yanqiu Shao, Ting Liu. *Improve Chinese Semantic Dependency Parsing via Syntactic Dependency Parsing*. In Proceedings of International Conference on Asian Language Processing 2012 (IALP 2012). 2012.10, pp. 53-56, Hanoi, Vietnam.

[8] **张梅山**, 车万翔, 刘挺, 邓知龙. 统计与词典相结合的领域自适应中文分词. 中文信息学报. 2012, 26(2):8-12.

# 哈尔滨工业大学学位论文原创性声明及使用授权说明

## 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《中文词法句法语义联合模型研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签名：　　日期：　　年　　月　　日

## 学位论文使用授权说明

本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，即：

（1）已获学位的研究生必须按学校规定提交学位论文；（2）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；（3）为教学和科研目的，学校可以将学位论文作为资料在图书馆及校园网上提供目录检索与阅览服务；（4）根据相关要求，向国家图书馆报送学位论文。

保密论文在解密后遵守此规定。

本人保证遵守上述规定。

作者签名：　　日期：　　年　　月　　日

导师签名：　　日期：　　年　　月　　日

# 致　谢

从2010年3月起正式加入哈工大社会计算与信息检索实验室成为其中的一员至现在一共四年半的时间，使我从一个自然语言处理中的新人也在这个领域的研究中有了少许的成果。实验室给我带来了很多收获，在这个大家庭里，感受到不少温暖。未来的日子里，我会继续努力，希望我将不是那个从实验室出去的成员中拖后腿的人，也希望以后继续能和实验室保持密切联系。

首先，我最感谢的是我的导师刘挺教授，在相处的日子里，刘老师不仅给了我不少作研究方面的建议，同时也在为人处事方面给了不少耐心的指导。我在很多方面都有所欠缺，以后我还需要根据刘老师的建议去逐渐提高我的综合素质。刘老师是我的良师益友，非常让人值得尊敬。在生活方面，刘老师也给予了我很大的支持。

然后，我非常感谢我们LA组的老大车万翔老师，他给予了我很多直接的指导工作。在每次的工作讨论中，车老师总是那么的随和，在我的每一篇论文中，都凝聚有他的智慧。平时，生活上经济上车老师也都对我非常关心，一起打球的日子也非常开心。

我也非常感谢李生教授、秦兵教授、张宇教授在博士的开题以及中期从大局上指出论文的不足之处。另外还要特别感谢秦老师在生活上的帮助，我非常能体会到这其中的温暖。此外，还要重点感谢新加坡科技与设计大学的张岳老师，他对于我论文的指导也有着非常大的作用，在考虑问题方面和写作逻辑表达方面都给出了很多具体的支持。

在实验室的师兄师弟师姐师妹中，首先特别感谢赵妍妍师姐、李正华师兄以及张伟男等等，妍妍师姐以及伟男在平时生活中的支持是让人难以忘怀的，李正华在启蒙论文中给予我的指导以及平时的学习生活交流给了我很大的帮助。其次，我也由衷的感谢帮我做过实验的师弟师妹们，尤其是包括邓知龙、刘一佳、丁宇以及王少磊等等。我也非常感谢实验室中所有陪我一起打羽毛球的人们，运动不仅是一种锻炼手段，也是一个自我心态调节的良好方式。最后感谢社会计算与信息检索研究中心的每一位成员，也希望每一位成员都会有更好的将来，希望这个大家庭不断壮大不断发展。

最后我非常感谢我的父亲、我的老婆以及我的二叔三叔，一切都不容易，有你们的支持以及在背后作出的默默牺牲才使得我这个博士能坚持下来。

# 个人简历

张梅山，男，1983年4月生，出生于湖北省松滋市。

## 教育经历

- 2010.3 – 现在：就读于哈尔滨工业大学计算机学院社会计算与信息检索研究中心，导师为刘挺教授。
- 2005.9 – 2008.7：就读于中国科学院软件研究所，获得工学硕士学位，导师为刘立祥副研究员。
- 2000.9 – 2004.7：就读于中国地质大学（武汉），获理学学士学位。

## 工作实习经历

- 2012.12 – 2013.3: 在新加坡科技与设计大学访问学习。在张岳助理教授指导下，研究了基于字的句法分析模型以及分词词性标注领域自适应问题，在自然语言处理领域相关会议上发表多篇论文。
- 2009.12 – 2010.3: 在北京百度公司的自然语言处理组实习，参与“招工工作”平台相关开发。
- 2008.5 – 2009.10: 在浙大网新集团工作，被外派在微软文本语音组参与软件开发的工作。

## 获奖情况

- 2007年度中国科学院软件研究所三好学生。
- 2001年获得2000年度中国大学生曾宪梓奖学金。
- 2001获得中国地质大学（武汉）校级三好学生。

## 参加评测

- 参与组织了NLP&CC 2013年的语义依存评测。
- 参加了2012年CIPS-SIGHAN的短语结构句法分析和微博分词评测。在短语结构句法分析方面，我们在提交的5个系统中获得了最好的性能；在微博分词评测方面，在提交的20个系统里面排名第二。
- 参加了2012年国际面向互联网数据的句法分析评测（SANCL），在提交的12个结果中，我们的系统排列第3名。
- 参与组织了SemEval-2012年的中文语义依存分析评测。