硕士学位论文

依存句法分析的置信度研究

RESEARCH ON CONFIDENCE MEASURE IN DEPENDENCY PARSING

郭 江

哈尔滨工业大学

2012 年 6 月

国内图书分类号：TP391.2　　学校代码：10213
国际图书分类号：681.37　　密级：公开

工学硕士学位论文

依存句法分析的置信度研究

硕士研究生：郭 江
导　　师：车万翔副教授
申 请 学 位：工学硕士
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：2012 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index: TP391.2
U.D.C: 681.37

Dissertation for the Master Degree in Engineering

RESEARCH ON CONFIDENCE MEASURE IN DEPENDENCY PARSING

Candidate: Guo Jiang
Supervisor: Associate Prof. Che Wanxiang
Academic Degree Applied for: Master of Engineering
Specialty: Computer Science and Technology
Affiliation: School of Computer Science and Technology
Date of Defence: June, 2012
Degree-Conferring-Institution: Harbin Institute of Technology

摘 要

句法分析是自然语言处理的核心问题，对信息抽取、信息检索、机器翻译等应用有重要的支撑作用。依存句法以其形式简洁、易于标注、便于应用等优点为人所重视。虽然目前依存句法分析算法研究取得了一定的进展，但是其准确率仍然不能满足实际应用的需要。针对这一问题，本课题并没有将全部研究重点放在提高依存句法分析的准确率上，而是提出了对依存句法分析结果进行置信度分析这一新问题，通过对依存句法树中每条依存弧的置信度进行计算，将置信度高的依存搭配用于特定应用系统，从而提高这些系统的准确率。

目前主要有两种主流的依存分析机制，一种是基于转移的依存分析，另一种是基于图的依存分析。针对这两种依存分析方法的不同，本文提出了相应的计算置信度的方法。基于转移的依存分析通过构造优化的转移序列来对句子进行解码，从而生成一棵依存句法树。其学习方式有局部学习（Local Learning）和全局学习（Global Learning）两种方式。针对局部学习的转移模型，本文提出基于似然概率以及基于重采样的置信度计算方法；针对全局学习的转移模型，本文提出基于K-Best结果投票的置信度计算方法。基于图的依存分析通过寻找最大生成树的方法获取一个句子的依存分析结果。其学习过程一般使用在线学习（Online Learning）算法。依存分析的图模型一般为线性模型，因此与基于转移的依存模型不同，其无法直接得到一条依存弧的似然概率。针对这一问题，本文首先利用图模型的输出近似估计依存弧的边缘概率，并以此作为该依存弧的置信度。同时，本文提出一种有指导的基于Logistic回归的置信度估计方法，该方法能够充分地利用更丰富的特征，并且通过有效的学习方式对特征进行赋权。

另外，本文还提出了若干种置信度估计的评价方法，一方面可用以评价置信度计算结果的优劣，另一方面也为置信度计算提供了优化目标。

最后，本文将含有置信度的依存句法分析结果应用于篇章级情感分析以及半指导依存句法分析，以观察其有效性。实验结果证明，对置信度信息的合理应用能够在一定程度上提高上层应用及依存句法分析自身的性能。

关键词： 依存句法分析；转移模型；图模型；置信度

Abstract

The syntactic parsing is the core issue of natural language processing. It can support lot of applications, such as information extraction, information retrieval, and machine translations. The dependency parsing, with its simple grammatical form, easy-tagging, and facilitate applications, have recently gained a wide interest. Although the dependency parsing has made some progress recently, its accuracy is still unable to meet the needs of practical application. In this thesis, we do not pay more attention to improve the accuracy of dependency parsing. We propose a novel natural language processing task: The Confidence Measure in Dependency Parsing. By computing the confidence of each arc in a dependency tree, we can only apply those highly confident arcs to particular practical applications. Thus the performance of these applications can be improved.

Practically all data-driven models that have been proposed for dependency parsing in recent years can be described as either graph-based or transition-based. In this thesis, we propose several confidence measure methods for these two sorts of models. In transition-based parsing, we learn a model for scoring transitions from one parse state to the next, conditioned on the parse history, and perform parsing by greedily taking the highest-scoring transition out of every parser state until we have derived a complete dependency tree. There are two learning algorithms for transition-based models, the Local Learning and Global Learning. For local learning models, we propose two methods of confidence measure. One is the Likelihood-based method, the other is Resampling-based method. For global learning models, we propose a Weighted K-Best Voting method which makes use of the K-Best outputs of the parser. In graph-based parsing, we instead learn a model for scoring possible dependency graphs for a given sentence, typically by factoring the graphs into their component arcs, and perform parsing by searching for the highest-scoring graph. Graph-based dependency parsing usually uses an online learning algorithm to learn a model, which is a discriminative linear model. Thus we're not able to achieve a probability/likelihood of an arc directly. To solve this problem, we propose a method which approximately estimates the marginal probability of an dependency arc. Then we use the marginal probability as the confidence value. Meanwhile, we propose a novel supervised algorithm, which is based on logistic regression to estimate the confidence. With this algorithm, more extra features can be exploited, and their weights be

learned automatically to achieve a state-of-the-art confidence measure system.

Furthermore, we propose several evaluation methods for confidence measure. They can be used not only as evaluation methods, but also can be the optimization objective of confidence computing.

Finally, we apply the confidence measure methods to two practical applications to see the effectiveness. One is the document-level sentiment analysis, and the other is a semi-supervised dependency parsing task. Experimental results show that the use of confidence information actually improves the performance of these applications.

Keywords: Dependency Parsing, Transition-based Model, Graph-based Model, Confidence

目 录

第 1 章 绪论

1.1 课题背景

在自然语言处理的研究中，对中文的处理一般分为由浅到深的几个过程，分别是：分词、词性标注、句法分析、语义分析。其中分词、词性标注技术相对而言已经非常成熟，在诸如信息抽取、文本分类、搜索引擎等许多应用中往往被作为文本处理的基础工作，从而被广泛使用。句法分析是对语言的较深层次的结构化分析，其基本任务是：针对一个句子，识别出其语法结构，并按照某特定的语法形式将其结构形式化地表示出来，是一种较为典型的结构预测任务。已有的针对句法分析的研究通常采用短语结构或者依存结构的语法形式。语义分析则又可以分为两个层次：浅层语义分析以及深层语义分析。浅层语义分析旨在识别出句子中的谓词以及相应的论元，以语义角色标注为代表；对深层语义分析的研究则很少，一是由于深层语义的表达形式难以确定；另外，不论是浅层语义分析或是深层语义分析，都立足于句法分析的基础之上。若句法分析技术得不到质的飞跃，语义分析将止步不前。

本文主要关注的是基于依存语法（Dependency Grammar）的句法分析，以及其置信度研究。依存结构的表现形式较之短语结构而言更为简洁，从而降低了对标注人员的要求，从应用角度来看，也更具普适性和实用价值。同时，也有观点认为，依存语法更符合人的思维形式。因此，近年来，越来越多自然语言处理领域的专家和学者转向依存句法分析的研究[1]。2006、2007连续两年的国际CoNLL(Computational Natural Language Learning)评测任务均是针对多语言的依存句法分析（包括汉语在内）[2][3]，2008年的评测则为依存分析与语义角色标注的联合任务[4][5]。许多国家和单位的研究者参与了该评测，并且很大程度上促进了依存句法分析技术的快速发展与成熟。

越来越多的应用需要对自然语言进行深层次的理解。搜索引擎需要更好地理解用户所输入的Query，才能够更全面地了解用户的需求或者意图，而句法分析是通往Query理解的必经之路；情感分析[6]中较为关键的子任务是抽取评价搭配，这种搭配信息往往是长距离搭配，需要依存结构对其进行刻画；再如自动问答[7]、机器翻译等任务，都依赖于句子的依存句法结构。已有的许多研究也表明，依存句法分析能够为这些应用带来有效的帮助[8][9]。

1.2 研究目的和意义

近几年来，依存句法分析的研究取得了较大的进展。目前最好的依存分析器在英文上的准确率已经达到了90%以上，在中文上也达到了80%以上。但是，这仅仅是在受限领域上的性能，比如新闻领域。而在依赖于句法分析的上层应用中，往往面临的是不受限领域的数据。针对这一问题，2012年，Google组织的SANCL评测“Parsing the Web”[1]中的一个关键任务就是利用大量未标注的数据来改善句法分析器在不同领域的分析性能。但是从评测结果来看，领域自适应的结果并不是很出色。

由此可见，对于实际应用而言，目前句法分析的结果并不完美，存在各种各样的错误。究其原因，一共有以下几点：

1. 错误级联。在实际应用中，对自然语言的处理往往需要先进行分词、词性标注等浅层分析；而分词、词性标注中的错误将对句法分析性能产生较大的影响。
2. 领域迁移。实际应用中所面对的领域数据往往不同于已有的句法分析器的训练数据。研究发现，当训练数据领域与测试数据领域不同时，会导致句法分析性能急剧下降。以英语为例，从华尔街日报树库（WSJ）移植到Brown 语料时，句法分析性能下降近8%[10]。这是制约依存句法分析或者一个自然语言处理系统实际应用的一个关键的问题。
3. 统计模型的固有泛化误差。目前的依存句法分析技术大都基于统计模型，是数据驱动的句法分析。而统计模型往往会受到自身能力的限制以及数据噪声的干扰。
4. 语言自身的歧义性。这些特性给句法分析带来一些难以避免的错误。

对于上层应用而言，这些句法分析的错误有时会导致错误的信息或者答案抽取结果。因此，一个很直观的想法是：如果在进行依存句法分析的同时，我们能够获知哪些句法分析结果是正确的，而其他的是不正确的，就能够在实际应用中进行恰当的取舍，如只使用正确的句法分析结果，而忽略错误的句法分析结果，从而进一步提高应用系统的准确率。例如：对于较为依赖句法分析结果的实体关系抽取系统（一种信息抽取）[11]，如果只抽取句法分析正确的句子中的实体关系，将能够获得更准确的实体关系库（提高准确率）。尽管这样做会丢失掉一部分实体关系，但是当我们面向基于互联网的大规模数据时，那些丢掉的实体关系极有可能从其他句法分析正确的句子中抽取出来。

[1]https://sites.google.com/site/sancl2012/home/shared-task

因此，如何识别依存句法分析结果是否正确成为一个新的关键问题。目前这一问题在句法分析领域还鲜有研究。本课题提出依存句法分析置信度分析的研究正是为了解决这一问题，使得依存句法分析器在对一个句子进行句法分析的同时，还能够给出其分析结果的置信度信息，从而使其在上层应用中更为“智能”和实用。

1.3 国内外研究现状

1.3.1 依存句法分析

依存句法分析是基于依存语法的一种自动句法分析方法。依存语法的提出可以追溯到50年以前[12][13][14][15][16]，然而自依存语法提出以来，却很少有针对依存句法分析的系统性的研究。直到近几年（近十年），依存语法才渐渐引起了学者们的研究兴趣。

依存语法的一个基本假设是认为句法结构本质上包含词和词之间的关系，这种关系称为依存关系。一个依存关系连接两个词，分别称为核心词和修饰词。同时，依存关系可以细分为不同的类型，以表示两个词之间的句法关系，如主谓关系（SBV）、动宾关系（VOB）等等。依存句法分析的任务是针对给定句子分析其依存语法结构。

形式化地，输入句子被表示为：$x = w_0 w_1 ... w_n$。w_i表示输入句子的第i个词（也可以认为包含了词和词性）。w_0是人工引入的一个虚拟节点，在依存结构中指向整个句子的核心词，即根节点。依存句法树表示为：$\mathbf{d} = \{(h, m, l) : 0 \le h \le n, 0 < m \le n, l \in \mathcal{L}\}$，其中$(h, m, l)$表示一条由$w_h$（父亲节点）指向$w_m$（儿子节点）的依存弧，依存关系为$l$；$\mathcal{L}$表示句法依存关系的集合。依存句法树应满足以下4个基本约束：

1. 单核心（Single-Head）：即句子中每一个词只能依存于一个核心词。
2. 弱连通（Weakly-Connected）
3. 无环（Acylic）
4. 无交叉弧（Non-Crossing），又称投影依存树（Projective Dependency Tree）。这一条件不是必须的，对于有些语言，如捷克语等，不满足这一条件。

图 1-1 为一个依存句法分析的示例，其中Root是虚拟节点，指向整个句子的核心。

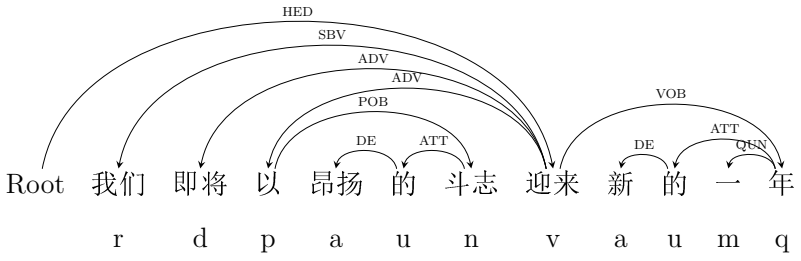

图 1-1 依存句法分析示例

依存句法分析的目的则是对于给定的输入句子**x**，寻找一棵得分最高的依存句法树d^*，即：

$$d^* = \arg\max_{\mathbf{d}} Score(\mathbf{x}, \mathbf{d}) \tag{1-1}$$

根据$Score$的定义不同，目前依存句法分析技术有两种主流的方法：

1) 基于图的方法

该方法将依存句法分析看成是在加权有向完全图中求解最大生树的问题[17]。基于动态规划的解码使得基于图的算法能够高效地在一个巨大的搜索空间中找到最优解，即最大生成树（Maximum Spanning Tree）。在基于图的方法中，依存句法树的分值被分解为多个子结构（子树）的分值之和：

$$Score(\mathbf{x}, d) = \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{d}) = \sum_{p \subseteq \mathbf{d}} Score(\mathbf{x}, p) \tag{1-2}$$

其中$\mathbf{f}(\mathbf{x}, \mathbf{d})$表示特征向量，**w**为相应的权值向量。$p$为分解后的子结构，可包含依存句法树$d$中的一条或者多条弧。当前模型中常用的子结构如图 1-2所示，子结构中包含的弧越多，模型阶数越高、越复杂，对应的解码算法复杂度也相应越高。

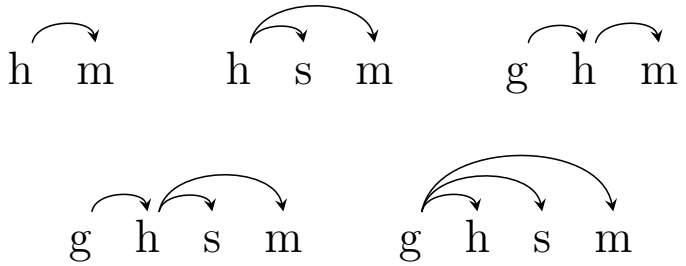

图 1-2 基于图的依存句法分析方法使用的子结构类型

2) 基于转移的方法

该方法将依存树的搜索过程转化为一个转移序列的构造过程，从而将依存分析问题转化为寻找最优转移动作序列的问题[18]。对于基于转移的依存分析模

型，其$Score$定义如下：

$$Score(\mathbf{x}, d) = \sum_{i=1}^{m} score(d_i|h_i) \tag{1-3}$$

其中，h_i表示前$i-1$个动作构成的转移历史，d_i表示根据当前历史采取的转移动作，如移进（Shift）、规约（Reduce）等。$score(d_i|h_i)$通常由一个分类器给出，如支持向量机，最大熵分类器等。

基于图和基于转移的方法近年来成为最主流的两种方法，并且都达到了较高的准确率。可以简单地从如下几方面对这两种方法进行对比。

1) 解码算法

基于转移的方法根据当前分析历史和分类器的结果，贪心地选择下一步转移动作。搜索过程无法回溯，因此错误级联比较严重。柱状搜索策略在一定程度上缓解了这种贪婪性。而基于图的方法通过动态规划，自底向上地从搜索空间中寻找分值最大的依存树，是一种精确解码算法。从解码效率上来看，基于转移的方法一般是$O(n)$[19]，而基于图的方法一般是$O(n^3)$[20]，因此基于转移的方法在实际应用中更有优势。

2) 学习算法

基于转移的方法有两种学习方式，一种是局部学习（Local Learning），一种是全局学习（Global Learning）。局部学习的目标是得到一个更准确的预测下一步转移动作的分类器；全局学习的目标则是得到一个更准确地预测整个转移序列的模型。基于图的方法的目标则是训练出一个能够让正确的依存树分值更高的模型。

3) 特征表示

基于转移的方法使用的特征更丰富，可以使用当前已经构建出来的所有子树信息，而基于图的方法由于要考虑到解码算法的效率，一般只使用局部的特征，如单条弧相关的特征、兄弟弧相关的特征，以及祖孙弧相关的特征等。

这两种方法侧重点不同，各有优缺点，从而也导致它们的分析结果的错误分布不同，因此可以相互弥补。近几年也出现了一些融合这两种模型的方法，如：后处理融合[21]、单向指导[22]、联合训练解码[23]等。

另外，由于依存句法分析标注数据的规模有限，领域覆盖面有限，而标注依存树库又费时费力。而与此同时，未标注数据却非常丰富，甚至可以认为是无限的。因此越来越多的研究者尝试利用未标注数据来提高依存句法分析的性能。如：利用词类特征[24]、利用自动分析结果中的子树[25]、利用双语对齐信

息[26]、自学习[27]以及大规模语言模型[28][29]等等。

1.3.2 依存句法分析置信度分析

通过以上介绍可以发现，目前依存句法分析的研究工作多集中于如何利用更好的算法或者更多的资源，来提高依存句法分析的准确率和效率。然而由于统计模型的泛化能力限制，以及自然语言固有的歧义性和不确定性，导致目前依存句法分析的结果并不理想。在CoNLL 2009 评测中，英语依存弧的准确率大概在87%，而整句的正确率只有40%。汉语相关指标则更低，依存弧的准确率大概在76%，而整句的正确率只有30%左右[5]。而随着领域的迁移，依存句法分析的性能则会受到更多未见数据的影响，从而准确率更低。上层应用系统若构建在含有较多错误的句法分析结果之上，其准确率必然会受到影响。为了解决这一问题，有一些学者开始研究依存句法分析的置信度问题。

Yates 等人（2006）最早对依存句法分析置信度问题进行研究[30]，他们利用语义分析的结果来检测错误的句法分析结果。Reichart 和Rappoport（2007）提出一种基于多个不同句法分析器的集成方法来选择高质量的句法分析结果[31]。Dell’Orletta 和Venturi（2011）利用某些启发式特征来检测可信的依存句法分析结果。Kawahara 和Uchimoto（2008）使用二元分类器，将句法分析结果分为两类：可信和不可信两类[32]。Ravi 等人（2008）采用线性回归预测句法分析器在一个句子集上的准确率[33]。然而，这些已有的研究工作关注的都是分析整个句子的句法分析结果的置信度。

尽管识别完整句子的句法分析结果的可信度能够被应用于主动学习[34]、半指导/无指导学习等方法[35]中，但是完整句子并非唯一的可信度度量对象，识别句法子结构的可信度同样是有用的。比如，有研究中使用从大规模自动句法分析的语料中抽取出来的子树或词对来帮助依存句法分析[25][36]。但是，这些子树或者词对的置信度仅仅由它们在语料库中出现的次数来决定。因此，他们的方法会对那些错误但是频繁出现的子树或者词对有所偏置，而忽视那些正确但是稀疏的子树或词对。所以有必要识别子结构（包括子树、句法路径、词对等）的可信度，从而为各种应用灵活的提供所需的信息。

1.4 本文研究内容

本文的研究内容依托于当前主流的两种依存分析技术：基于转移的方法与基于图的方法，并围绕依存句法分析的置信度计算方法以及置信度的应用两个

问题进行展开。在进行置信度计算之前，我们首先确定置信度分析的对象，接着分别针对基于转移的依存分析模型与基于图的依存分析模型提出了一系列置信度的计算方法，同时，也提出了相应的评价准则。最后，我们将依存句法分析的置信度信息应用于一种信息抽取任务——篇章级情感倾向性分析，以及一种半指导依存句法分析方法中，以检验其有效性。

本文各章节安排如下：

第一章，首先介绍了本课题的研究背景，接着阐述了研究目的与意义；然后分别从依存句法分析、依存句法置信度分析以及依存句法分析的领域移植三个方面综述了当前国内外的研究现状。最后，在此基础之上提出了本文的主要研究内容。

第二章，首先介绍了依存句法分析的相关理论，接着依托于当前主流的两种依存分析技术：基于转移的依存分析与基于图的依存分析，分别从解码算法及学习算法两个方面对依存句法分析进行深入地探讨。最后在此基础之上提出置信度分析的问题，并且确定本文的置信度分析对象。

第三章，针对基于转移的依存句法分析的特点，提出了三种计算依存弧置信度的方法，分别是基于似然概率的方法、基于重采样的方法以及基于K-Best依存分析结果的方法。同时，为不同的方法设计了相应的实验，并且采用两种评价方法对这三种置信度计算方法进行对比。

第四章，针对基于图的依存句法分析的特点，提出了两种计算依存弧置信度的方法，分别是基于近似边缘概率的方法以及基于Logistic回归的方法。基于Logistic回归的方法是一种有指导的置信度计算方法，模型的本质是对依存弧进行二元分类，并且在分类的同时给出置信度信息。同时，为不同的方法设计了相应的实验，并分别进行评价和对比。

第五章，为了验证依存句法分析置信度在实际应用中的价值，首先将句法树的置信度信息应用于篇章级情感分析，篇章级情感分析是一种信息抽取的应用，其主要任务是针对一条评论（电影、酒店、书等），为其打上情感标签，如褒义、贬义等。同时，还将置信度信息应用于依存句法分析自身，尝试在大规模未标注数据中抽取出置信度较高的句法分析信息，来帮助提高依存句法分析的性能。这种利用未标注数据改进模型的方法一般称为半指导的方法。

第 2 章 依存句法分析

2.1 句法分析

句法分析基于分词以及词性标注之上，是一种深层次的自然语言处理技术。其基本任务是对于一个自然语言句子，挖掘出其内部的语法结构，并将其形式化。句法分析的主要目的是解决自然语言中的歧义问题。我们知道，自然语言中存在着各种各样的歧义，一个句子可能是多种语义的共同载体。一般来讲，对于中文而言，有三种歧义在日常生活中比较常见：

1. 分词歧义。对于同样一个句子，不同的分词结果可表达不同的意思。例如：南京市长江大桥。这句话（短语）有两种分词方式：(1). 南京市/长江大桥；(2). 南京/市长/江大桥。这两种分词方式表达的语义截然不同。再如：“白天鹅在水中游”可理解为“白天/鹅/在/水中/游”，也可理解为“白天鹅/在/水中/游”。这种歧义现象在中文分词的研究中屡见不鲜。
2. 词义歧义。即使分词的方案只有一种，也可能产生不同的意思。例如：门没有锁。这个句子的分词结果只有一种：“门/没有/锁”，但是由于“锁”既有可能是动词，也可能是名词，因而使得整句话产生了不同的意思。
3. 句法歧义。对于一个句子而言，即使它的分词方案唯一，每个词的词义也没有变化，整个句子仍然可以表达不同的涵义。例如：咬死猎人的狗。这句话的语义既可以指“把猎人的狗咬死了”，也可以指“一只咬死了猎人的狗”。

在英语中同样会存在类似的问题，但是汉语因其更为灵活多变，从而导致歧义的现象更为频繁。

句法分析最主要的目的便是为了解决自然语言中句法层次上的歧义问题。

2.1.1 依存语法

短语结构语法中有终结点与非终结点之分，刻画的是句子的产生过程，即认为：一句话（词序列）是在一系列产生式规则所构成的集合的驱动之下，一步一步产生的。而依存语法则不然。实际上，我们很难描述一句话究竟是如何产生的，尤其对于汉语而言。汉语是一种意合语言，对词序以及词的完整性要

求往往不高。在这种情形之下，依存语法则更为合适。依存语法并不试图回溯到“句子是如何产生的？”这个深刻的命题，而旨在对已产生的句子，分析出其词与词之间相互依存的关系，并形成一棵完整的树状结构。

在依存语法中，我们通过有向的依存弧来表达两个词（或子结构）之间的依存关系。同时，为体现具体的句法结构类型，每条依存弧上都附着一个标记，该标记称为依存关系。例如，在句子“我/要/毕业/了”所对应的依存结构中，“毕业”是“我”的谓语（动作），“我”则是“毕业”的主语，两者之间构成主谓关系，因此，在该依存结构中，存在一条由“毕业”指向“我”的有向弧，弧上关系为主谓（SBV, SuBject Verb）。其中，依存弧的发出者称为该依存搭配的核心词，也称支配词；依存弧箭头的指向者称为该依存搭配的修饰词，也称从属词。

由前所述，句法分析的目标是对自然语言进行句法层次上的消歧，那么，依存语法是如何实现该目标的呢？以句子：“咬死猎人的狗”为例，通过依存句法分析，分别可以得到两种语法结构，如图 2-1所示：

图 2-1 ”咬死猎人的狗”的两种依存句法结构

图中的两种依存结构则表征两种不同的语义。在第一种结构中，“咬死”作为整个句子的核心节点（不考虑Root），为一个缺省主语的谓词。“咬死”的宾语是“狗”，“猎人”则通过“的”与“狗”之间构成定语修饰关系。在第二种结构中，“狗”是整个句子的核心节点，前面的所有词均为它的修饰成分。对同一个句子的两种理解方式则通过两种不同的结构表现出来，同时，根据依存句法分析模型，我们可以得到以上两棵依存句法树的分值或者概率，然后根据该分值或者概率，选择最优的结构，从而实现句法结构上的消歧。

本文的研究内容基于依存句法分析进行展开。

2.1.2 数据驱动的依存句法分析

早期的句法分析研究中，人们一度采用基于规则的方法，也称基于语法的方法（Grammar-based）。基于规则的句法分析器根据语言学家所建立的形式系统（语法）所建立。对于任一个输入的文字串（句子），根据制定好的语法即可推导出其语法结构。在此方法中，语言学家参与的比重比较大。

然而对于基于规则的方法而言，有两个问题很难克服。首先，规则的获取完全依赖于语言专家的知识以及经验，从而导致开发规则的代价巨大。另外，由于人的知识表示能力还存在不足，无论多么复杂的一个形式系统都无法涵盖所有的语言现象。实际上，哥德尔不完全性定理也表达了类似的思想：即不存在一个形式系统，能够表示自然界中所有的“真”。在语言学中，这预示着不存在这样一个语法体系，它能够推导出所有在真实世界中合理的自然语言句子。

从上世纪90年代开始，基于统计的自然语言处理技术渐渐吸引了人们的注意。一方面是由于机器学习理论和技术的逐渐完善；另一方面，语料资源获取的门槛越来越低。近几年兴起的众包（Crowd Sourcing）技术更是为大规模数据采集提供了难得的契机。从机器学习的角度来看，依存句法分析实质上一个结构预测的问题，与分类、回归问题有所不同。因此，依存句法分析的研究也为结构预测的理论提供了一个非常好的研究场景。基于统计的依存句法分析通常也被称为数据驱动（data-driven）的依存句法分析。

基于转移的依存分析与基于图的依存分析是两种主流的数据驱动的依存分析方法。接下来，将分别从解码以及学习两个角度，对这两种方法进行比较和分析。

2.2 基于转移的依存分析算法

基于转移（Transition-based）的依存分析算法是一种确定性的分析算法[37]。该算法将句子的解码过程建模为一个转移序列的构造过程。其依存分析模型的目标是通过学习得到一个能够准确的预测下一步转移动作的分类器。

2.2.1 解码算法

基于转移的依存分析系统中，每一个转移状态可以表示为一个三元组：$<\sigma,\beta,A>$，其中σ表示一个栈，用以存储当前已经部分处理过的词；β则表示尚未处理过的词构成的序列；A为在当前状态下的部分依存树，即当前已经建立的依存弧集合。算法2-1描述了基于转移的解码过程。

给定一个输入句子W，句法分析器的初始状态为$<nil,W,\Phi>$，终止状态为$<S,nil,A>$。最终得到的是一棵弱连通、无环、无交叉、单父节点的树。Nivre提出了两种转移算法，分别为arc-standard算法和arc-eager算法。对于这两种转移算法，分别有不同的操作类型与之对应。

Input: Sentence W

Initialize: $c \leftarrow < nil, W, \Phi >$

Termination: $c_t \leftarrow < S, nil, A >$

while $c \neq c_t$ **do**

  $t \leftarrow \lambda_c$

  $c \leftarrow t(c)$

end

return $G(c)$

算法 2-1 基于转移的依存分析解码算法

对于arc-eager算法，有以下四种转移类型：

1. 左弧规约（Left-Arc，以下简称LA）。给定当前状态为$< w_i|\sigma, w_j|\beta, A >$，其中，$w_i$为栈顶词，$w_j$为$\beta$中下一个待处理词。此操作将生成一条由$w_j$指向$w_i$的依存弧，依存关系为$r$，并将该依存弧添加至当前生成树中，同时弹出栈顶词$w_i$。执行该操作的后继状态为$< \sigma, w_j|\beta, A \bigcup \{(w_j, r, w_i)\} >$。在进行LA操作时，要求当前状态下的部分依存树$A$中不存在$(w_k, r', w_i)$，即$w_i$不能同时依存于两个节点（单父亲节点约束）。
2. 右弧规约（Right-Arc，以下简称RA）。给定当前状态为$< w_i|\sigma, w_j|\beta, A >$，此操作将生成一条由$w_i$指向$w_j$的依存弧，依存关系为$r$，并将该依存弧添加至当前生成树中，同时，将$w_j$压入栈中。执行该操作的结果状态为$< w_j|w_i|\sigma, \beta, A \bigcup \{(w_i, r, w_j)\} >$。
3. 规约（Reduce，以下简称RE）。给定当前状态为$< w_i|\sigma, \beta, A >$，此操作从栈中弹出元素w_i。
4. 移进（Shift，以下简称SH）。给定当前状态为$< \sigma, w_i|\beta, A >$，移进操作将w_i压入栈。

对于arc-standard算法，则有三种转移类型：

1. 左弧规约（LA）。此操作与arc-eager算法中的LA操作类似，不同之处在于，这里无需判断当前状态下的依存树A中是否存在依存弧(w_k, r', w_i)，在arc-standard算法中，只要节点尚在栈中，其一定还没有找到父亲节点。
2. 右弧规约（RA）。与arc-eager算法不同，给定当前状态：$< w_i|\sigma, w_j|\beta, A >$，此处RA操作在建立一条由$w_i$指向$w_j$的依存弧之后，不将$w_j$压入栈中，而是将栈顶元素$w_i$替换掉待处理词序列头部元素$w_j$，同时将$w_i$从栈顶弹

出。

3. 移进（SH）。与arc-eager算法类似，在给定当前状态为$S, w_i|I, A$的情况下，Shift操作将w_i压入栈。

2008年，Yue Zhang提出基于柱搜索（Beam-Search）的转移算法，较大优化了基于转移的句法分析性能。在传统的基于转移的解码算法中，给定当前分析器的状态s，其下一步转移动作由以下式子进行预测：

$$T(s) = \arg\max_{T \in ACTION} Score(T, s) \tag{2-1}$$

其中：$ACTION = \{SH, LA, RA, RE\}$。那么这种方式的缺点很明显，就是贪心选择所带来的错误级联。一旦在转移过程中出现错误的预测，则会影响接下来的一系列转移动作。一个很直观的想法就是在进行每一步转移时，考虑所有转移操作，这些操作将会产生多个后继状态。利用柱搜索的思想，再从所有的后继状态中选择最优的K个状态（K为柱的宽度），以进行后续转移，直到解码完成。假设对于句子x，最终我们得到的候选依存树集合为$GEN(x)$，则最优的结果可由以下式子获得：

$$F(x) = \arg\max_{y \in GEN(x)} \sum_{T' \in act(y)} Score(T', s_{T'}) \tag{2-2}$$

式中，T'为解码得到依存树y的转移序列中的一步转移动作。$s_{T'}$为执行转移动作T'时，分析器的状态。

2.2.2 学习算法

对于传统的贪心搜索策略下的转移算法，通常采用分类的方式进行转移动作的决策。这种学习方式针对每一步转移进行优化，将执行每一步转移动作时的上下文和操作类型进行组合，作为分类器学习的一个实例。也称局部优化。实际上，任意一种多元分类算法都可应用于此。例如支持向量机（SVM），最大熵，和一些在线学习算法（如MIRA, PA等）。分类学习算法的目标是学习一个映射（决策）函数：$f : Configuration \rightarrow \{LA, RA, RE, SH\} \times \{R \bigcup nil\}$，其中$R$表示依存关系集合。RE(Reduce)和SH(Shift)无需与依存关系进行组合。

对基于柱搜索的转移算法，Yue Zhang采用线性模型计算每个转移动作的得分[23]：

$$Score(T, s) = \Phi(T, s) \cdot \vec{w} \tag{2-3}$$

式中$\Phi(T, s)$为根据当前状态以及转移动作类型所抽取的特征向量。模型的目标是学习得到优化的权值向量$\vec{w}$。Yue Zhang采用泛化的感知器算法[38]训练模型。

与Nivre的局部优化方法不同，对基于柱搜索的转移模型，可以进行全局优化。其学习算法伪代码如下：

Input: training samples (x_i, y_i)
Output: $\vec{w}$
Initialize: set $\vec{w} = 0$
// R training iterations; N examples
for *t=1 to R, i=1 to N* **do**
    $z_i = \arg\max_{y \in GEN(x_i)} \sum_{T' \in act(y_i)} \Phi(T', c') \cdot \vec{w}$
    if $z_i \neq y_i$ **then**
        $\vec{w} = \vec{w} + \sum_{T' \in act(y_i)} \Phi(T', c_{T'}) - \sum_{T' \in act(z_i)} \Phi(T', c_{T'})$
    end
end

算法 2-2 基于转移的依存分析的感知器学习算法

2.3 基于图的依存分析算法

基于图（Graph-based）的算法将依存分析转换为在有向完全图中求解最大生成树的问题，是基于动态规划的一种图搜索算法。该算法由McDonald于2005年提出[39]，是一种全局最优的搜索算法。

对于句子：$x = w_1 w_2 ... w_n$，首先构建一个完全有向图$G = (V, E)$，其中V表示句子中的所有词（包含引入的虚拟词w_0），E为图中有向边的集合，需要注意的是，在图G中，两个词（即节点）之间可以有多条同方向的边，以表示不同的依存关系。基于图的方法则将依存分析的过程视为在该有向图G中，求解最大生成树的问题。

2.3.1 解码算法

McDonald采用Eisner算法[40]作为依存分析的解码算法。该算法于1996年由Eisner提出，并以此命名。

Eisner算法的伪代码如算法2-3所示。

其中，$C[s][t][\leftarrow][0]$表示从s到t的一个不完整span（Incomplete Span）；←表示该span的方向，这里则表示t为该span的核心节点；0表示这个span是不完整的，

```
1  Initialize: C[s][s][d][c] = 0.0, ∀s, d, c
2  for k=1 to n do
3      for s=1 to n do
4          t = s + k
5          if t > n then
6              break
7          end
8          # Create incomplete items
9          C[s][t][←][0] = max_{s≤r≤t} C[s][r][→][1] + C[r + 1][t][←][1] + score(t, s)
10         C[s][t][→][0] = max_{s≤r≤t} C[s][r][→][1] + C[r + 1][t][←][1] + score(s, t)
11         # Create complete items
12         C[s][t][←][1] = max_{s≤r≤t} C[s][r][←][1] + C[r + 1][t][←][0]
13         C[s][t][→][1] = max_{s≤r≤t} C[s][r][→][0] + C[r + 1][t][→][1]
14     end
15 end
```

算法 2-3 Eisner算法伪码

即s的左侧还存在以s为核心节点的其他节点，也就是s的子孙节点。$score(t, s)$表示以t为核心节点，s为依存节点的依存弧的分值或者概率。

在Eisner算法的自底向上搜索的过程中，在两个完整span组合成一个不完整span时，建立一条依存弧（算法2-3步骤9、10）；而在由完整span与不完整span合并成一个完整span过程中，不建立依存弧。

基于上述算法的依存模型在计算一条依存弧的分值时，只利用与该弧相关的特征，因此，也被称为一阶模型（first-order graph-based model）或者弧分解模型（Arc-factored model）。一阶模型隐含的假设是：依存弧之间是相互独立的。而该假设的约束显然过强。因此，McDonald于2006年提出了二阶模型（second-order）[17]，在一阶特征的基础之上，引入了与兄弟节点相关的特征。在二阶模型的基础之上，Carreras于2007年提出了高阶模型[41]，使用了三阶的祖孙节点相关的特征。

引入高阶特征之后，上述基于span的Eisner算法便不再具有动态规划所要求的优化子问题特性了，即：两个span合并之后得到的span分值最高（最优解），并不意味着两个子span也是最优的，还取决于第三部分的分值。针对这个问题，

通常有两种办法，一种是通过柱状搜索的策略来近似最优解，李正华（2009）采用了这种方法[42]，利用所有孙子节点构成祖孙特征。另一种是动态规划，Terry Koo(2010)[20]为三阶模型设计了新的优化子结构，并提出了高效的高阶动态规划解码算法，时间复杂度为$O(n^4)$。

2.3.2 学习算法

基于图的依存分析模型的训练通常采用Online Learning算法。Online Learning算法的一般形式如算法2-4所示。

Input: Training data: $T = \{(x_t, y_t)\}_{t=1}^{T}$

1 $w_0 = 0; v = 0; i = 0$
2 **for** *n=1 to N* **do**
3 **for** *t=1 to T* **do**
4 $w^{(i+1)}$ =update $w^{(i)}$ according to instance (x_t, y_t)
5 $v = v + w^{(i+1)}$
6 $i = i + 1$
7 **end**
8 **end**
9 $w = v/(N * T)$

算法 2-4 Online Learning算法框架

Online Learning算法的核心在于第4步，即权值的更新。根据更新的准则不同，有不同的Online Learning算法。目前常用的Online Learning算法主要有Perceptron（感知器）、AP(Averaged Perceptron)[43]、PA(Passive-Agressive)算法以及K-Best MIRA(K-Best Margin Infused Relaxed Algorithm)[44].如前所述，在基于柱搜索的转移算法中，我们用到了Perceptron；在基于图的依存分析模型中，通常采用的是K-Best MIRA更新准则，如下式：

$$\begin{aligned} &\min \|\mathbf{w}^{(i+1)} - \mathbf{w}^{(i)}\| \\ &s.t. \forall y' \in dt_{k-best}(x_t), score(x_t, y_t) - score(x_t, y') \le Loss(y_t, y') \end{aligned} \tag{2-4}$$

其中：x_t是第t个训练实例（句子），y_t为训练语料中句子x_t所对应的正确句法树，又称gold-tree。$dt_{k-best}(x_t)$表示由当前的模型（权值向量）解码所得到的K-Best依存句法树。$score(x_t, y_t)$为根据当前模型计算得到的正确依存树y_t的分

值（或者概率），$score(x_t, y')$则为依存树y'的分值（或者概率）。$Loss(.)$为损失函数，$Loss(y_t, y')$为y'的损失值。$Loss(.)$可以有很多种选择，通常在依存句法中使用Haiming Loss，即0-1 Loss，定义如下：

$$Loss(y_t, y') = arc\_err(y_t, y') + label\_err(y_t, y') \tag{2-5}$$

K-Best MIRA的更新准则涵义是：使得对于K-Best预测结果中的任意一棵依存树，其损失越大，那么在调整模型（更新权值）之后，它与正确句法树之间的间隔（Margin）也应越大。并且在满足这个约束的条件下，要尽可能小地调整当前的特征权值向量。

2.4 融合模型

从理论上进行分析，基于图的依存分析与基于转移的依存分析有很多的不同。基于图的依存模型是一个全局优化的模型，采用精确解码（动态规划）。其缺点是只能利用有限的句法分析历史。而基于转移的依存模型则恰恰相反，它是一个局部优化的模型，采用贪心算法进行解码。其优点是在进行转移决策时，能够利用更丰富的特征，包括决策历史。

有研究表明，对于基于图的依存分析模型与基于转移的依存分析模型，两者的错误分布有很大的不同[45]。这种差异给两者的融合带来了契机。

Nivre和McDonald提出一种Stacking的方式对两种模型进行融合[22]。这种融合方式也称基于特征的融合（Feature-Based Integration）。主要思想是由一种模型的分析结果来为另一种模型提供指导特征（Guided features）。实验证明，这种融合方式使得两种模型的性能都取得了较大的提高。值得一提的是，这种Stacking的融合方法也被应用于分词、词性标注等任务，且都取得了较好的结果[46]。

Yue Zhang提出联合解码的方法对两者进行融合[23]。在他的方法中，采用基于Beam-Search的Covington算法[47]和arc-eager算法分别作为基于图和基于转移的的依存分析解码算法。两者具有诸多相似之处：都是自左向右的递增式分析、都是线性模型、且都采用全局学习（Global Learning），也都利用柱搜索进行解码。因此在计算一棵依存树的分值时，只需要将两个模型的打分函数相加，便得到融合模型下的打分函数。利用该打分函数，在Beam-Search的框架之下，便可实现联合解码。

2.5 依存句法分析的置信度

依存句法的置信度分析是一个比较新的课题，也是本文的主要研究内容。通过1.2节的描述，我们可以看到，置信度分析对于实际应用的意义是非常大的。置信度分析的通常做法是将整个句子作为置信度评价的对象，即判断一个句子的句法分析结果正确与否。然而，若想得到一个句子完全正确的句法树（所谓完全正确，是指该依存句法树中的每一条弧，或者每一个子结构都分析正确）是很难的。CoNLL 2009年的评测结果显示，只有约30%的中文句子和约40%的英文句子能够被完全正确地句法分析。另外，如果只关注整棵句法树的置信度，会带来以下问题：

1. 一棵被判别为可信的句法树（其置信度较高）中仍然可能存在一些错误的句法子结构。而对于不同应用，其所需要的关键子结构是不同的。例如，对于语义角色标注（Semantic Role Labeling）任务[48]而言，句法树的主干结构（bone structure）是关键子结构；而对于多词表达（Multiword Expression）任务[49]，枝干结构（branch structure）则是关键子结构。假如关键子结构分析错误，那么即使整棵句法树的置信度较高，对应用而言也没有帮助，甚至是负作用，从而导致较低的精确率（Precision）。
2. 一棵被判别为不可信的句法树（其置信度较低）中可能包含一些分析正确的句法子结构。以实体关系抽取为例，只要句中两个实体间的句法路径分析正确的话，我们就可以得到正确的实体关系。因此，简单地丢掉被判别为不可信的句法分析结果将会导致低的召回率（Recall）。

因此，除了将整棵句法分析树作为置信度分析的对象之外，我们还可以更细致地分析句法树中依存弧、句法路径、依存子树等结构的置信度。

本文将依存弧作为置信度分析的基本对象。

2.6 本章小结

本章首先从自然语言中常见的三种歧义现象出发，介绍了句法分析的基本任务。接着针对当前比较受亲睐的依存语法以及在依存语法支配下的依存句法分析进行了分析与介绍。同时引出了数据驱动的依存句法分析。

然后，针对当前主流的两种数据驱动的依存句法分析技术：基于转移和基于图的依存句法分析方法，分别进行了详细的探讨。探讨主要从两个角度出发，一个是模型的解码算法、另一个是模型的学习算法。

接下来，针对两种模型的差异，引出了对融合模型的思考。并且针对当前已有的且被证明有效的融合方法进行了简要的分析。

最后，分析了当前已有的依存句法分析置信度研究，探讨了将整个句子作为置信度分析对象的缺点，以及会带来的问题。同时，阐述了对更细粒度的依存分析结果进行置信度分析的必要性，并提出了本文进行置信度分析的对象，即依存弧。

第 3 章 基于转移的依存句法分析置信度计算

3.1 基于似然概率的置信度分析

通过之前的介绍，我们知道，基于转移的依存句法分析主要通过四种转移动作来完成，分别为Left-Arc(LA)，Right-Arc(RA)，Reduce(RE)，Shift(SH)。尽管对于Arc-standard算法与Arc-Eager算法而言，四种转移类型所执行的动作并不尽相同，但是它们有一个共同点：在执行LA/RA操作时，将建立一条向左指向或是向右指向的依存弧；而对于RE/SH操作，不建立依存弧。因此，一个非常直观的想法是，在基于转移的依存分析解码过程中，将执行LA/RA操作的概率作为根据该操作所建立的依存弧的置信度。

由此，我们使用最大熵模型来训练基于转移的依存分析器。解码过程如算法3-1所示。最大熵模型有着悠久的历史，是一种基于信息论的对随机事件概率分布进行预测的建模方法。通过简单的数学分析可以得到，根据最大熵模型计算得到的条件概率分布正好是Log-linear形式的似然概率（见算法3-1步骤4）。

Input: Sentence W

1 Initialize: $c \leftarrow < nil, W, \Phi >$

2 Termination: $c_t \leftarrow < S, nil, A >$

3 **while** $c \neq c_t$ **do**

4 $\quad t^* \leftarrow \arg\max_{t \in \mathcal{T}} P(t|x)$ where $P(t|x) = \frac{\exp(\mathbf{w} \cdot f(x,t))}{\sum_{t' \in \mathcal{T}} \exp(\mathbf{w} \cdot f(x,t'))}$

5 $\quad c \leftarrow t^*(c)$

6 $\quad$ **if** $t^* = LA \ \ or \ \ RA$ **then**

7 $\quad\quad Confidence(new\_arc) = P(t^*|x)$

8 $\quad$ **end**

9 **end**

10 return $G(c)$

算法 3-1 利用最大熵模型预测下一步转移动作

在以上算法第6步，对t^*进行判断，若$t^* = LA$ 或 RA，则将$P(t^*|x)$作为规约所建立的依存弧的置信度。

3.2 基于重采样的置信度分析

Mejer 2010采用基于重采样的方法[50]，对词性标注任务进行置信度分析，其分析的基本单位是词性标注中的每一个词。重采样方法基于贝叶斯观点，认为模型中每一维特征所对应的权值应服从某一个概率分布，而不是一个固定的值。在此假设之下，我们对现有模型按照某种概率分布进行K次重采样（对每一维特征的权值分别重采样），便可以得到K个新的模型。接下来利用这K个新的模型对句子重新解码，从而得到K个新的预测结果。最后，使用这K个预测结果，对原模型的预测结果进行投票，“赞成票”所占总票数比例即可作为置信度。

权值的概率分布也称为模型先验，通常采用高斯先验（Gauss Prior）。即认为每一维特征所对应的权值服从某一高斯分布。根据该高斯分布的方差选择方式不同，有两种不同的重采样方式，分别称为KD-Fix(K-Draws by Fixed Standard Deviation)和KD-PC(K-Draws by Parameter Confidence)。顾名思义，前者假设所有权值符合同一高斯分布，即高斯分布的参数对于所有权值都是一致的；后者则是通过参数估计的方法获得每一维特征相应的高斯分布参数。

受此启发，我们在依存句法分析中，同样设计了两种重采样方式，并依此进行置信度的计算。

3.2.1 KD-Fix

利用最大熵模型进行分类时的决策函数如下式：

$$P(t|x;\mathbf{w}) = \frac{\exp(\mathbf{w}\cdot f(x,t))}{\sum_{t'\in\mathcal{T}}\exp(\mathbf{w}\cdot f(x,t'))} \tag{3-1}$$

在KD-Fix方法中，我们假设$\mathbf{w}$服从高斯分布$\mathbf{w}\sim\mathcal{N}(\mu,\Sigma)$。其中$\mu$为当前模型的权值，$\Sigma$为协方差矩阵。这里，再作进一步假设：令各维特征之间相互独立，此时协方差矩阵Σ是一个对角阵，其非对角线元素为0。

接下来，根据该高斯分布对每一维权值进行K次重采样，从而得到K个采样之后的最大熵模型。利用这K个新的模型对测试数据（句子）分别进行K次预测，便可以得到K棵依存句法树。最后，再使用这K棵句法树对原模型所预测的依存句法结果进行投票。假设$\mathbf{z}^{(i)}(i=1...K)$为对输入句子$\mathbf{x}$的K种预测结果，$\hat{\mathbf{y}}$是真实模型的预测结果。那么，对于句中的某个词$p=1...|\mathbf{x}|$的依存弧$\hat{y}_p$，其置信

度可由下式计算：

$$Confidence(p) = |\{i : \hat{y}_p = z_p^i\}| / K \tag{3-2}$$

另外，在KD-Fix中，高斯分布的方差如何选择将是需要解决的一个问题。

3.2.2 KD-PC

在KD-Fix中，我们简单地假设线性模型中特征的权值服从某高斯分布，并依此对它进行采样。这是一种非常启发式的方法，由于我们缺少对模型中每一维权值的认知，因此只能假设所有权值服从同一个高斯分布。那么我们能否在模型学习的过程中，融入这种先验知识，从而获得每一维权值具体的分布参数呢？Crammer于2008年提出Confidence-Weighted Learning，正是这样一种融合高斯先验的Online学习算法[51]。

由于传统的Online Learning算法存在对稀疏特征的权重学习不充分的情况，导致模型中稀疏特征所对应的权值往往置信度不高。因此在Confidence-Weighted Learning中，为每一维特征引入高斯分布的假设：假设模型参数（权值向量）服从期望为μ，协方差矩阵为Σ的高斯分布。如果我们假设各维特征之间相互独立的话，这里的协方差矩阵将是一个对角阵，对角阵的每一个对角元素则是每一维权值的方差。而每一维权值的方差同时也反映了该权值的置信程度。权值的方差越大，表示该权值在学习过程中更新不平稳，因而其置信度较低。反之，则该权值的置信度较高。因而，在算法迭代的过程中，可以根据某权值方差（置信度）的大小调整更新的步长。一般来讲，置信度高的权值更新要尽量小（Conservative），而置信度低的权值要尽量大地更新（Agressive）。与传统的Online Learning算法不同的是，Confidence-Weighted Learning的模型参数并非特征权值，而是每一维权值的期望以及方差。期望值可以作为基于该模型进行预测时的特征权重，而方差则为该权重的置信度。

因此，与KD-Fix方法不同，在KD-PC中，我们可以根据模型自身进行重采样。

3.3 基于K-Best依存分析结果的置信度分析

在基于柱搜索的转移模型中，我们很容易得到一个句子的K-Best分析结果。同时，K-Best结果所对应的分值，实际上反映了整棵依存树的一种相对置信度信息（$Score \notin [0,1]$）。因此，在本方法中，我们利用句子的K-Best分析结果及

其相应得分，采用加权投票的方法，来近似地计算一条依存弧的置信度。

假设依存句法分析的K-Best结果的分值分别为$w_i(i = 1...K)$，那么，对于句中某个词$p = 1... | \mathbf{x} |$的依存弧$\hat{y}_p$，其置信度计算方式如下：

$$Confidence(p) = \frac{\sum_{i\ s.t.\hat{y}_p=z_p^i} Score_i}{\sum_i Score_i} \quad (3\text{-}3)$$

其中，$Score_i = \max\{0, \mathbf{w} \cdot \Phi(x, z^{(i)})\}$

3.4 实验结果与分析

3.4.1 实验数据

我们使用哈工大社会计算与信息检索研究中心自主标注的中文依存句法树库CDT(Chinese Dependency Treebank)[52]作为本实验数据。CDT一共包含来自人民日报的6万个句子，详细数据信息以及本实验所采用的数据划分情况如表4-1所示。其中依存弧数目为总词数中去掉标点符号之后的数目。

表 3-1 CDT数据统计

	句子数	总词数	平均每句词数	依存弧数目
训练集	55,496	1,026,054	18.5	878,968
开发集	1,500	29,091	19.4	24,870
测试集	3,000	56,786	18.9	48,486

对基于似然概率的方法，我们在训练集上训练基于转移的依存分析模型，在开发集上进行调参，最后在测试集上进行测试。我们修改其解码算法，使得其在进行句法分析的同时，也输出相应的置信度信息。

对基于重采样的KD-Fix方法，我们在已经训练好的依存分析模型基础之上进行K次重采样。在本实验中，K被设置为50。这里需要注意的是，进行高斯采样时的标准差σ是一个可调的参数，我们在开发集上对其进行调优。

对基于K-Best依存分析结果的方法，我们采用的是Yue Zhang的基于柱搜索的依存分析器，取K为50。

3.4.2 评价方法

对每一种置信度计算的方法，我们都根据置信度值将所有的依存弧进行分组。在本实验中，我们一共划分了20个区间，每个区间的长度为0.05。对于每

一个置信度区间，我们分别计算在该区间内依存弧的精确率，即被预测正确的依存弧所占的比例。那么在理想的情况下，这个比例应该与该置信度区间的中值相等。

形式化地，假设置信度在$[(j-1)/20, j/20), j = 1...20$范围内的依存弧被分到第$j$个区间，令$b_j$为第$j$个区间的中值，即：$b_j = j/20 - 1/40$。同时，区间$j$里的依存弧（置信度$v \in [(j-1)/20, j/20)$）中被预测正确的比例（精确率）为$c_j$，理想情况下应当有：$b_j = c_j$。$c_j$可以认为是实际准确率，而$b_j$则为期望的准确率。假如一种置信度估计的方法的结果是$c_j < b_j$，则说明该方法对置信度的估计过于悲观，未达到期望；反之，若$c_j > b_j$，则说明该方法对置信度的估计过于乐观，高于期望值。

由此，我们通过“实际准确率-期望准确率”曲线作为置信度分析结果的评价方法。其横轴为期望准确率，即b值；纵轴为实际准确率，即c值。

同时，我们也采用均方根误差（root mean-square error，简称RMSE）作为另一种评价机制。RMSE由下式进行计算：

$$rmse = \sqrt{(\sum_j n_j(b_j - c_j)^2)/(\sum_j n_j)} \tag{3-4}$$

式中，n_j为第j个区间中依存弧的数目。

3.4.3 结果与分析

我们以CDT测试集中每条依存弧作为置信度分析对象（不考虑标点符号），分别采用上述四种置信度估计的方法进行分析。同时，以最直观的基于似然概率的方法作为Baseline方法进行比较，再利用“期望-实际（准确率）”曲线以及均方根误差两种评价方式对这四种方法进行比较。同时，我们也给出随机置信度下的结果作为参考。为方便表示，我们采用以下记号：

- ME-LL (Maxent Likelihood)：基于似然概率的方法。
- KD-Fix：基于固定标准差的高斯重采样方法。
- KD-PC：基于Confidence-Weighted Learning的高斯重采样方法。
- WKB (Weighted K-Best)：基于K-Best依存分析结果的加权投票方法。

性能对比如图3-1所示：

依存句法分析器的性能一般使用LAS(Labeled Attachment Score)以及UAS(Unlabeled Attachment Score)进行评价。LAS和UAS分别定义如下：

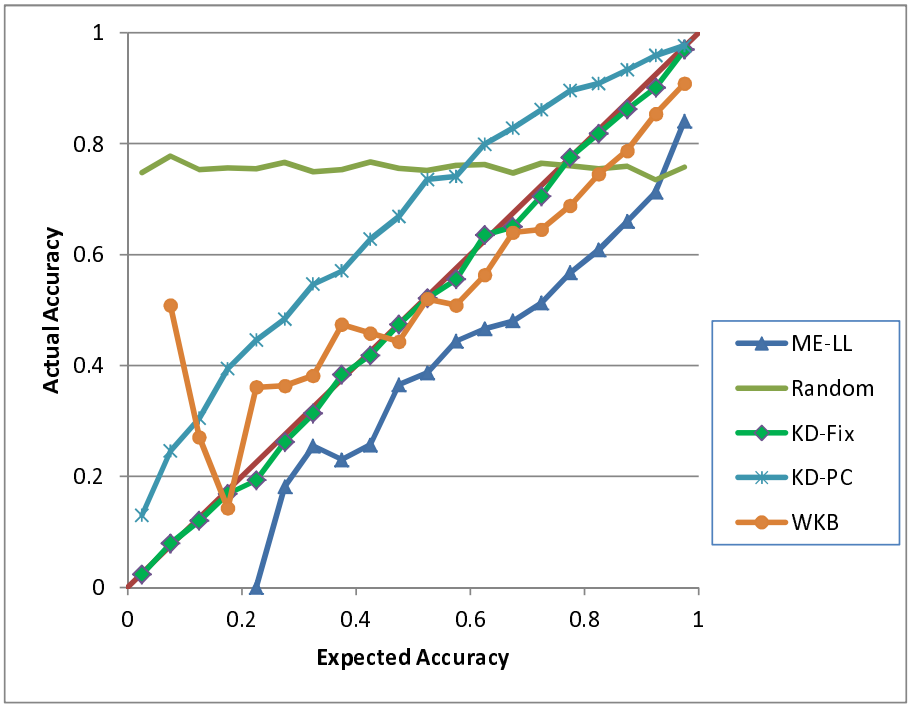

图 3-1 ME-Likelihood, KD-Fix, KD-PC, WKB分别在CDT上的期望-实际准确率曲线

$$LAS = \frac{\text{弧正确并且依存关系正确的词数}}{\text{所有词数}} \times 100\%$$
$$UAS = \frac{\text{弧正确的词数}}{\text{所有词数}} \times 100\% \quad (3\text{-}5)$$

从图3-1中，我们首先通过Random曲线可以大致看出依存分析器的LAS值，约为76%。由于Random方法是随机给依存弧赋予置信度，因此每个区间的实际准确率（Actual Accuracy）分布较为均匀。

作为Baseline方法，ME-Likelihood曲线在理想曲线$y = x$的下方，说明，采用似然概率作为置信度的估计方法过于乐观。固定标准差下的高斯重采样方法能够很好地拟合$y = x$，证明了这种基于模型重采样的投票方法的有效性。需要说明的是，我们在CDT开发集上对标准差进行了细致的调参，最终标准差$\sigma = 0.3$。同时，基于Confidence-Weighted Learning的KD-PC方法却没有取得非常好的效果，从拟合程度上来看，KD-PC曲线与ME-Likelihood曲线类似，不同之处在于，ME-Likelihood曲线在$y = x$下方，表明对置信度的估计过于乐观，而KD-PC则恰恰相反。另外一个值得注意的地方是，KD-PC曲线相对ME-Likelihood曲线而言，更为平滑，变化趋势更为稳定。

WKB方法从与曲线$y = x$的拟合程度上来看，优于ME-Likelihood与KD-PC方法。同时，在低置信度区间（< 0.7），WKB曲线的局部波动较为明显。而在高

置信度区间（> 0.7）则更为平滑和稳定。

期望-实际准确率曲线的波动与每个区间内依存弧的总数有关。由于依存分析器的整体性能在70% ~ 80%之间，因此处于高置信度区间内的依存弧数目较多，从而得到的实际准确率也更为准确。我们同时采用了RMSE对这四种方法进行比较，如图3-2所示。可以看到，RMSE反映了类似的评价结果。

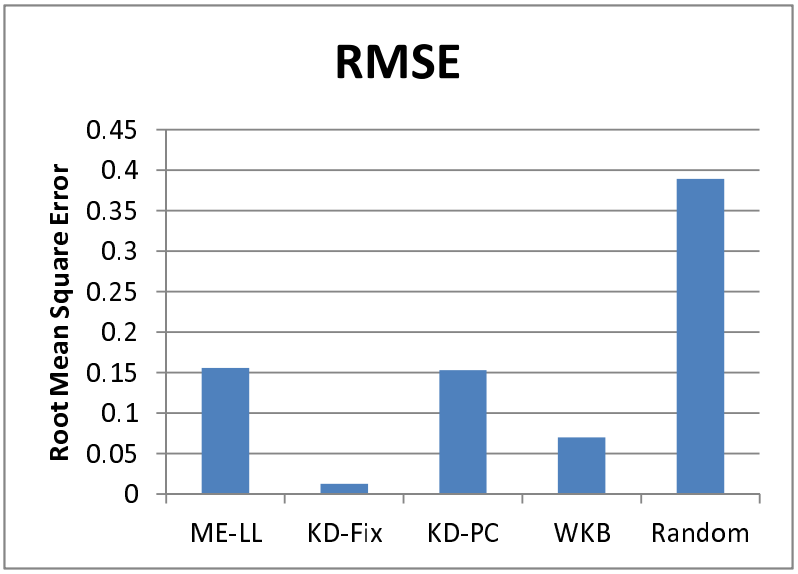

图 3-2 ME-Likelihood, KD-Fix, KD-PC, WKB分别在CDT上的RMSE值

3.5 本章小结

本章从基于转移的依存句法分析模型出发，提出了三类计算依存弧置信度的方法。第一类方法最直观，也最易实现，是基于似然概率的方法。第二类方法稍微复杂一点，是基于模型重采样的方法。对模型进行重采样的基本假设是，模型中每一维特征的权值服从某一高斯分布。根据对该高斯分布的标准差进行不同的设定，又有两种不同的重采样方法。其一是按照固定标准差的高斯分布进行重采样，即KD-Fix(K-Draws by Fixed Standard Deviation)；另外一种基于Confidence-Weighted Learning，根据模型学习过程中所得到的各维权值的方差进行重采样。对于KD-Fix方法，由于存在可调参数，即高斯分布的标准差，因此，我们在CDT的开发集上对其进行调优。第三类方法基于依存分析器所产生的K-Best句法树，对待估计依存弧进行加权投票。在该方法中，我们采用基于Beam-Search的转移模型获得K-Best依存分析结果。

最后，我们在CDT(Chinese Dependency Treebank)语料上对这三类方法分别进行了实验和对比。实验中，我们采用的评价方式有两种：一是“实际准确

率-期望准确率”曲线，通过观察其与理想曲线$y = x$的拟合程度，来评价该方法的优劣。另外一种评价准则是均方根误差（root mean-square error）。

通过实验，我们发现，无论从准确率曲线的拟合程度还是均方根误差来看，KD-Fix方法的效果比其他三种方法（ME-Likelihood，KD-PC，WKB）性能都要好很多。而ME-Likelihood与KD-PC方法的均方根误差非常接近，但是ME-Likelihood的估计过于乐观，即对置信度的估计过高；而KD-PC恰恰相反，对置信度的估计过低。但是相对而言，KD-PC曲线更为稳定和平滑。WKB方法比ME-Likelihood和KD-PC要好，但是曲线波动较大，不太稳定。

第 4 章 基于图的依存句法分析置信度计算

4.1 基于近似边缘概率的置信度分析

通过第二章的介绍，我们知道，基于图的依存分析是一种判别式的线性模型，而不是生成式的概率模型。在这种判别式模型中，我们只能得到每条依存弧的分值（Score），而不能获得它的准确概率。因此，本文借鉴(Koo et al. 2007)的思想[53]，首先采用一种近似的方法来估计依存弧的边缘概率，并以此作为Baseline方法。

在估计依存弧的概率之前，我们首先计算一棵依存树的概率，计算方法如下：

$$P(d|x;M)=\frac{\exp(Score(x,d))}{Z(x;M)} \tag{4-1}$$

$$\text{其中，}Z(x;M)=\sum_{d'\in\mathcal{T}(x)}\exp(Score(x,d')) \tag{4-2}$$

其中，M表示依存分析模型，$Z(x;M)$为归一化因子。$\mathcal{T}(x)$为句子x的所有可能依存句法树集合。

得到依存句法树的条件概率$P(d|x;M)$之后，其任一条依存弧(h,m,l)的概率可由下式进行计算：

$$P((h,m,l)|x;M)=\sum_{d'\in\mathcal{T}(x):(h,m,l)\in d'}P(d'|x;M) \tag{4-3}$$

需要注意的是，在计算依存句法树以及依存弧的条件概率时，都需要对集合$\mathcal{T}(x)$中所有依存树的分值进行求和或者遍历。然而$\mathcal{T}(x)$是关于句子长度n的指数级大小。为了简化计算，本文采用基于图的依存模型的K-Best依存分析结果对$\mathcal{T}(x)$进行近似。其中K被设置为1000。

4.2 基于Logistic回归的置信度分析

我们知道，在基于图的依存句法分析模型中，受解码的时间复杂度所限，往往只能利用较为有限特征，比如单条依存弧特征、兄弟特征以及祖孙特征（高阶）等。在本节中，我们提出一种有指导的置信度分析方法。在该方法中，我们可以利用更多的特征，用以辅助依存弧的置信度判断。

在一棵依存句法树中，我们把所有依存弧分为两类：正类（父亲节点与儿子节点，以及依存关系都是正确的）和负类（其他情况）。于是，可以很自然地将依存弧的置信度分析看作一个二元分类问题，或称可信依存弧识别。对于分类问题，一般需要解决四个子问题：

1. 使用什么分类器?
2. 使用什么特征?
3. 训练数据如何构造?
4. 如何评价分类结果?

与一般的二元分类问题不同，对于可信依存弧识别问题而言，正负样本是不平衡的。当前最好的英文依存句法分析器的LAS（Labeled Attachment Score）达到了90%，这意味着正确预测的依存弧与错误预测的依存弧的比例达到了9:1。在中文上，当前最好的中文依存句法分析器在CDT数据上的LAS值约为81%。正样本与负样本的比例约为4:1，不平衡性比在英文上稍微弱一点。

因此，在解决上述问题的过程中，我们需要考虑类样本不平衡的问题。

4.2.1 分类器：Logistic回归

本文采用Logistic回归训练分类器[54]。之所以采用Logistic回归，主要原因是Logistic回归是一种概率模型，能够估计出样本属于某个类别的概率。而这个概率，可以直接作为依存弧的置信度。另外，Logistic回归拥有高效的训练和预测速度。本文采用的是L2正则化的Logistic回归，即在Logistic回归的优化目标基础之上，进行L2正则化，以减小结构风险，防止过拟合（Overfitting）。其模型优化的目标函数如下：

$$\begin{aligned}\arg\min_{w} \frac{||w||^2}{2} &+ C^{+} \sum_{\{i|y_i=+1\}}^{n_+} log(1+e^{(-y_i w^T x_i)}) \\ &+ C^{-} \sum_{\{i|y_i=-1\}}^{n_-} log(1+e^{(-y_i w^T x_i)})\end{aligned} \tag{4-4}$$

其中(x_i, y_i)表示第i个训练样本，w为权值向量。C^+与C^-分别为正类和负类对于分类错误的惩罚因子，也是控制结构风险与经验风险之间的折衷的一个参数。在学习Logistic回归模型时，通过将C^-设置得大一些，可以在一定程度上防止分类器将负样本错分为正样本。

预测时，Logistic回归通过下式计算样本(x_t, z_t)属于正类的概率，从而对其进行分类：

$$P(y_t = +1|x_t; \mathbf{w}) = \frac{1}{1 + \exp(-y_t \cdot \mathbf{w} \cdot x_t)} \tag{4-5}$$

其中，x_t表示由依存弧所抽取出的特征向量，若该依存弧被正确标注，则$z_t = +1$；反之则$z_t = -1$。条件概率$P(y_t = +1|x_t; \mathbf{w})$则表示依存弧被正确标注的概率，因此可以作为该依存弧的置信度。

4.2.2 特征

对一个分类问题而言，特征的选择几乎决定了分类器的性能。对于可信依存弧的识别问题，我们为之定义了如下三类特征：

1. 文本相关的特征（Text-based features）。此类特征只与该句子的原始文本相关，而与其依存分析结果无关。
    - 句子的长度。从目前句法分析器的特点以及性能来看，长句往往更难以分析正确，由此我们认为，句子长度将影响依存弧的分析。句子越长，正确的几率越低。由于句子的长度是一个实数值，因此在本文的实验中，我们将其离散化至三维特征：**LS**(long-sentence，句子长度大于40)，**MS**(middle-sentence，句子长度介于16与40之间)，以及**SS**(short-sentence，句子长度小于15)。LS/MS/SS对于句子长度的划分是在开发集上调参的结果。
    - 未登录词（OOV）的数目。未登录词是指从未出现在训练数据中的词语。未登录词越多，句子被分析正确的概率越低。由于一句话中未登录词数往往不多，因此，本实验中，我们未将其作进一步的离散化，而是简单地将未登录词数当作离散特征处理。

    需要注意的是，前两类特征可视作一个句子的全局特征。

对该句子中的所有词，这两类特征值都是相同的。然而，一个句子中通常情况下既有正确预测的依存弧，也有错误预测的依存弧，所以，这两类特征也会给分类器引入一定的干扰。同时，我们也利用了一些基础特征：

- 当前词是否未登录词。
- 当前词的词型：w_m，m为当词前的下标。
- 词性Bigram：$t_{m-1}t_m$以及t_mt_{m+1}。
- 词性Trigram：$t_{m-1}t_mt_{m+1}$。

2. 句法相关的特征（Parser-based features）。此类特征与依存句法分析结果相关。有以下几类：
   - 依存弧长度：即当前词w_m与其父亲节点w_h之间的距离。依存弧长度是影响弧准确率的重要因素。
   - 依存词搭配：w_mw_h，w_h为w_m的父节点。
   - 依存词词性搭配：t_mt_h。
   - 依存关系类型：l_m。在本实验中，我们同时考虑了依存弧的方向，但是实验结果证明弧方向帮助不大。
   - 当前词词性及依存关系搭配：t_ml_m。
   - 中间词特征（InBetween features，词性的Trigram形式）：$t_mt_bt_h$，其中，t_b表示句中位于m和h之间的词的词性。
   - 环绕词特征（Surrounding features，词性的4-gram形式）：$t_{m-1}t_mt_ht_{h-1}$，$t_{m-1}t_mt_ht_{h+1}$，$t_{m+1}t_mt_ht_{h-1}$，$t_{m+1}t_mt_ht_{h+1}$。
3. 句法分析器的一致性特征（Agreement）。一个很直观地想法是，假如两个不同的依存句法分析器对某依存弧的标注是一致的，那么该依存弧的可信程度比较高。我们使用基于转移的依存句法分析器作为参考。对同一数据，分别使用两个依存句法分析器对其进行分析，假如两个句法分析器对一条依存弧标注一致，则对这条弧而言，该特征值为True，反之则为False。

4.2.3 训练数据的构建

本文采用N折交叉验证的方式构造可信依存弧识别系统的训练数据。其构造以及训练流程如图 4-1所示：步骤①将CDT训练集均匀地分为N份，每一次选

择其中的N-1份训练模型，剩下的一份作为测试集（步骤②）。该过程重复N次，就可以得到整个依存句法训练集上的自动句法分析结果。通过和正确的句法树进行比较，将所有依存弧分为正负两类，至此，可信依存弧识别的训练语料便构造完成（步骤③）。最后，在构建好的训练数据上训练可信依存弧识别模型（步骤④）。

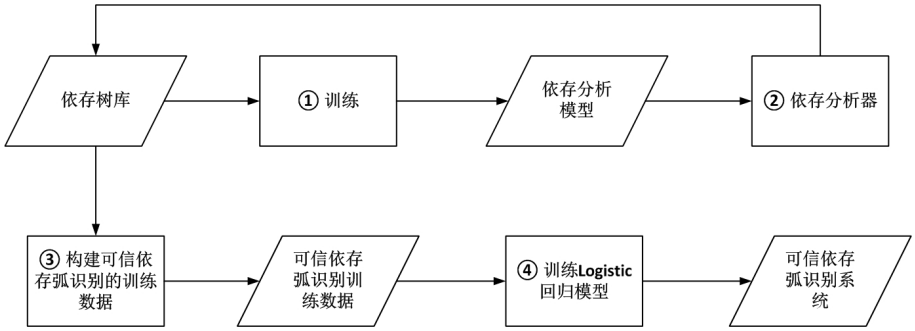

图 4-1 可信依存弧识别系统的训练过程

4.2.4 评价方法

(1) ROC曲线与PR曲线

对分类问题而言，通常采用准确率（Accuracy）作为评价指标。然而，根据之前的分析我们看到，在可信依存弧识别的问题中存在类样本不平衡的问题，在英文中这种不平衡性尤其明显。在类样本不平衡的情况下，准确率是否还适用呢? 举一个简单的例子：假设世界上99.9%的人不患癌症，只有0.01%的人身患癌症。导师交给我们一个任务，让我们设计一个分类器，判断某病人是否身患癌症。有一个小伙子非常聪明，想到一种“绝妙”的方法：既然99.9%的人不患癌症，那么在这个先验知识基础之上，只需要让分类器将所有样本（病人）判别为不患癌症，那么准确率即可达到99.9%。这是一个非常高的准确率，但是显而易见，该分类器没有任何实用价值。同理，对于n:1的类样本分布，若简单地将所有样本决策为大类，便可获得较高的准确率，然而，这并不能反映分类器的实际性能。

因此，在类样本不平衡的情况下，准确率并非一种合适的评价标准。本文采用两种在不平衡的情形下更为准确的两种评价方法：ROC(Receiver Operating Characteristic)曲线以及PR(Precision-Recall)曲线。

ROC曲线是基于混淆矩阵的一种评价方式。以二元分类器为研究对象，下面的混淆矩阵显示了一个分类器可能遇到的所有情况：

矩阵中的列对应于样本实际类别（gold），行对应于样本被预测的类

表 4-1 混淆矩阵（Confusion Matrix）

	positive (gold)	negative (gold)
positive' (predict)	TP(true positive)	FP(false positive)
negative' (predict)	FN(false negative)	TN(true negative)

别（predict）。若样本实际类别为positive，而被预测为negative，则用TP(True Positive)来表示，其他同理。由矩阵中四个基本指标可以衍生出多种分类器的指标。

- $FPrate(FPR) = FP/N$。N为负样本数。
- $TPrate(TPR) = TP/P$。P为正样本数。
- $Accuracy = (TP + TN)/(P + N)$。
- $Precision = TP/(TP + FP)$。
- $Recall = TP/P$
- $F1 = 2 \cdot Precision \cdot Recall/(Precision + Recall)$

其中准确率(Accuracy)是最常用的评价方式，根据之前的论述，在类样本不平衡的情形，准确率会导致有误导性的结论。ROC曲线描述的是FPR与TPR两个量之间的相对变化关系。在Logistic回归模型中，我们可以得到样本属于某个类别的概率，那么通过设定不同的分类阈值，可以得到不同的混淆矩阵。而每一个混淆矩阵则对应于ROC曲线上的一个点。由这些点描绘出来的，便称为ROC曲线。图4-2 为一个ROC曲线的例子，其横轴为FPR，纵轴为TPR。若混淆矩阵所对应的点处在对角线$y = x$之上，表明该分类器与随机猜测的性能相当，正常情况下，ROC曲线应当处于对角线的上方。最后，我们可以通过计算ROC曲线下面积（AUC, Area Under the Curve）作为对分类器性能的综合评价。

同时，也有研究表明，在类样本不平衡性较为显著的情况下，PR(Precision-Recall)曲线比ROC曲线更能反映分类器的实际性能[55]。PR曲线以召回率（Recall）为横轴，精确率（Precision）为纵轴。本文同时采用ROC曲线与PR曲线作为评价方式。

(2) P@N曲线

在实际应用中，前N条最可信的依存弧是比较重要的，此时，我们将可信依存弧识别的问题视作一个排序问题，可以采用信息检索领域常用的P@N进行评价。对于不同的应用，N的取值并不相同。例如，在搜索引擎中，由于用户

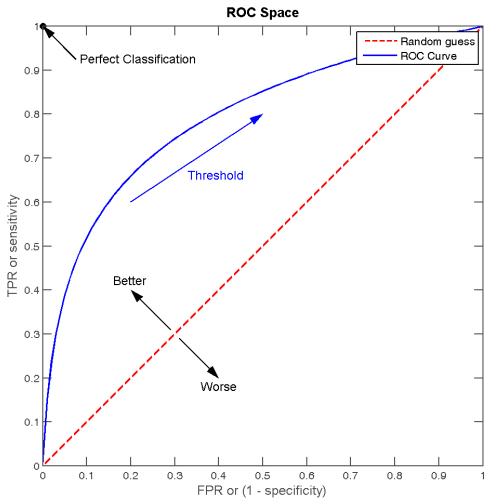

图 4-2 ROC曲线示例

在通常情况下只点击前面的搜索结果，因此N一般取为10或者20。但是对于可信依存弧的识别问题，N的值将会比较大；尽管如此，在不同应用中，N的取值也不尽相同。因此，在对可信依存弧识别的评价中，我们针对不同的N值绘制了P@N曲线，以更直观的方式展现其性能。

4.3 实验结果与分析

4.3.1 实验数据

我们使用CDT(Chinese Dependency Treebank)作为本实验的数据集（参考3.4.1节）。同时，使用开源的Mate-tools[1]作为依存分析器。Mate-tools实现了基于动态规划的高阶依存分析模型，通过Hash Kernel对特征进行降维，并且将句子间的处理并行化，是一个高速而且高性能的依存分析器[56]。我们对其源码进行了简单的修改，使之输出K-Best依存分析结果，以估计依存弧的近似边缘概率。另外，由于基于Logistic回归的方法是以分类问题的视角对依存弧的置信度进行建模，因此，我们给出CDT上的依存分析准确率，以观察其类不平衡情况，如表4-2所示：

从表4-2可以看出，正样本数目为负样本数目的4 ~ 5倍左右。需要注意的是，训练集的依存分析性能是4折交叉验证的结果。

[1]http://code.google.com/p/mate-tools/

表 4-2 CDT上的依存句法分析性能

	LAS(%)	正例:负例
训练集	81.37	4.37 : 1
开发集	81.19	4.32 : 1
测试集	81.43	4.39 : 1

4.3.2 结果与分析

4.3.2.1 特征贡献

为检验4.2.2节中定义的三类特征分别的贡献，我们首先使用所有三类特征训练分类器，然后，再分别去掉某一类特征，并观察分类器性能的下降。从下降的幅度可以看出哪一类特征贡献度更大。表4-3为在开发集上进行检验的结果（以AUC-PR为优化目标）。另外，通过在开发集上的调参，最终Logistic回归的参数设置是：$C^{+}=1, C^{-}=1.4$。意味着对于错误分类的负例，将会给予更大的惩罚。因此，这种设置可以在一定程度上防止负样本被错误分类。

表 4-3 CDT开发集：不同类特征的贡献

特征类型	AUC-PR(%)	性能下降（%）
All Features	94.89	N/A
−Text-based Features	94.81	0.07
−Parser-based Features	93.06	1.75
−Agreement Feature	93.90	0.99

从表4-3我们可以看出，这三类特征对可信依存弧的识别都有所贡献，但是各有不同。文本相关的特征贡献最小（0.07%），经分析，原因很可能是文本相关的特征并不考虑句法结果，从而只是考量该依存弧或句子被分析正确的难度。而这只是一种非常弱的参考。同时，像句子长度特征、句中OOV数目特征均为句子级的全局特征，对依存弧没有区分，从而也引入了一定的噪声。另外，句法相关的特征以及不同句法分析器间的一致性特征对分类都有较显著的贡献。

4.3.2.2 可信依存弧识别结果

表 4-4 为在CDT测试集上，基于近似边缘概率的置信度估计以及基于Logistic回归的方法的性能对比。从AUC-PR以及AUC-ROC值来看，两种方法都取得了比较好的效果，其中基于Logistic回归的方法优于近似边缘概率的方

法。ROC曲线（图 4-3）、PR曲线（图 4-4）与P@N曲线（图 4-5）分别提供了更为直观的对比。从两个图中，我们可以得到一致的结论。

表 4-4 CDT测试集上的AUC-PR与AUC-ROC评价结果

评价准则	Logistic回归	近似边缘概率
AUC-PR	**94.65%**	93.59%
AUC-ROC	**81.42%**	79.38%

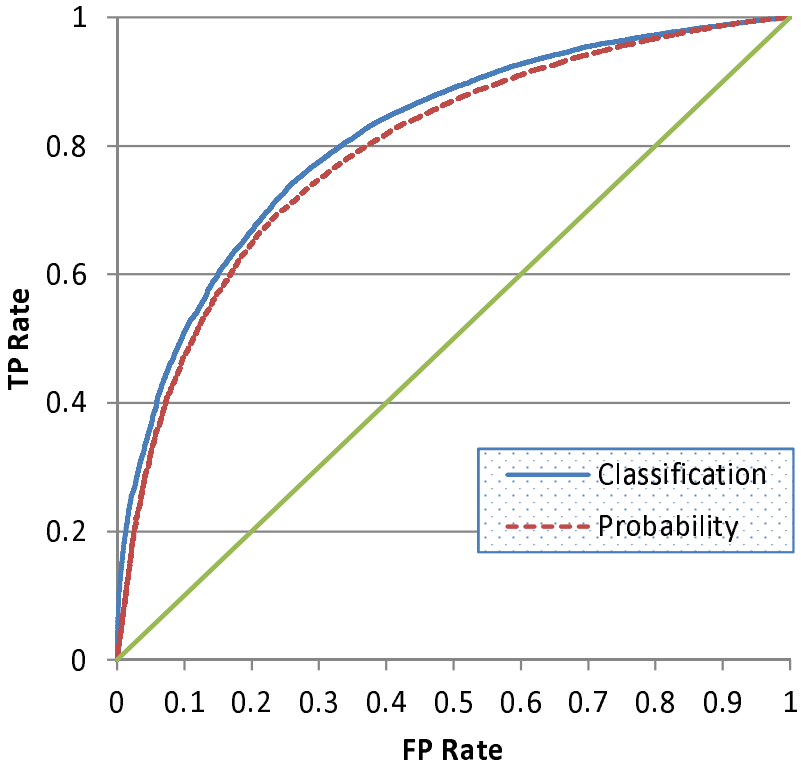

图 4-3 Logistic回归与近似边缘概率方法的ROC曲线

实际上，当K取无穷大时，近似边缘概率的方法可以得到更优的置信度估计，但是这也带来了非常高的计算代价。在基于图的依存分析解码过程中，当K的取值很大时，不仅耗时更长，而且内存占用率往往很高，普通机器通常难以承受。相对而言，基于Logistic回归的方法以较低的计算代价取得了更好的估计效果。通过分析，我们认为基于Logistic回归的方法之所以效果更好，原因主要在于其充分利用了更多的额外信息，如不同句法分析器之间的一致性。实际上，这借鉴于融合模型的思想。同时，该方法能够通过一种有效的学习机制为各个特征赋权。

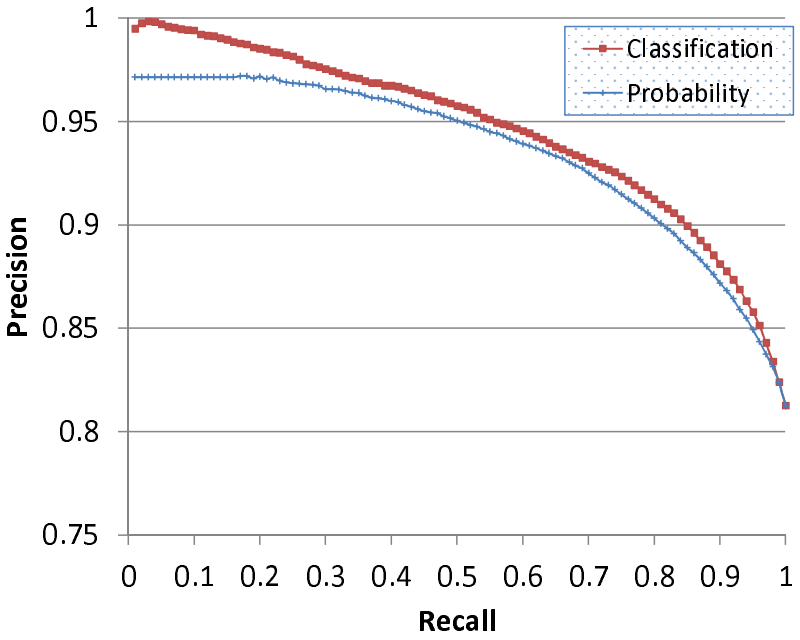

图 4-4 Logistic回归与近似边缘概率方法的Precision-Recall曲线

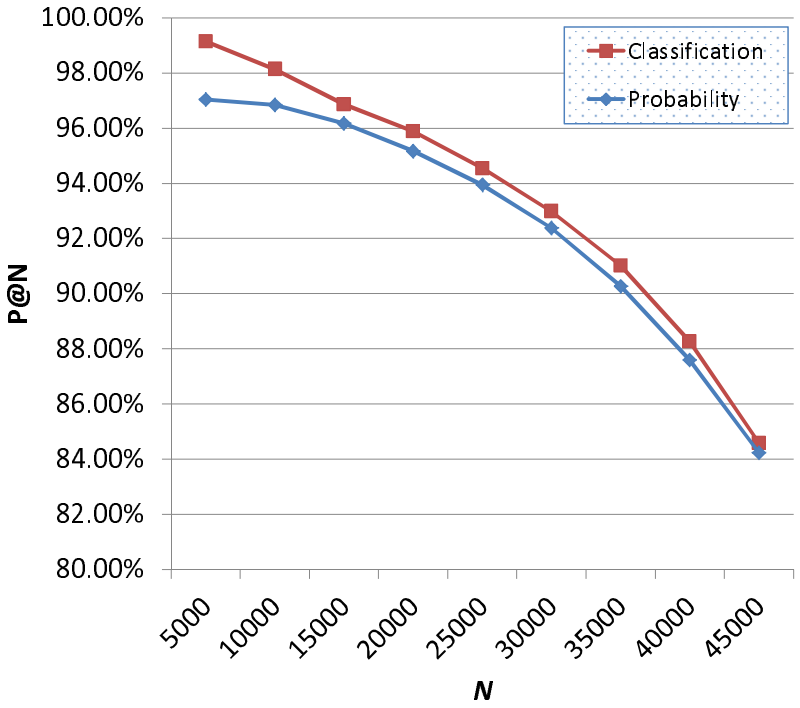

图 4-5 Logistic回归与近似边缘概率方法的P@N曲线

4.4 本章小结

本章从基于图的依存句法分析模型出发，针对模型特点，提出了两种计算依存弧置信度的方法。

第一种方法中，置信度是通过近似计算每条依存弧的边缘概率而获得的。该方法在线性模型的基础之上做了对数-线性模型假设，是一种启发式的方法。该方法的性能较严重地依赖于K的取值大小。理想情况下，K的大小为句子长度N的指数级。作为一种近似的估计，在本文的实验中，取K为1000。

第二种方法是一种有指导的方法，从分类问题的视角来看待依存弧的置信度分析。于是，问题转换为一个基于二元分类的可信依存弧识别问题。而对于分类问题而言，通常需要解决四个问题：1.分类器；2.特征选择；3.数据；4.评价方法。本章针对可信依存弧识别问题的特点，分别对这四个问题进行了分析和阐述。首先，本文选择L2正则化的Logistic回归算法训练分类器。之所以选择Logistic回归，首先因为它是一种概率模型，在分类的同时能够估计样本属于某个类别的概率，而该概率可以作为置信度。其次，Logistic回归拥有高效的训练和预测速度。接下来，我们为可信依存弧识别问题设计了3大类特征，分别为文本相关的特征、句法相关的特征以及多个依存句法分析器之间的一致性特征。然后本章描述了训练数据的构建过程，并针对该问题中存在的类样本分布不平衡的性质，提出了三种适用的评价方法。

最后，我们在中文依存树库（CDT）上对这两种方法进行了实验和对比。首先通过特征“去皮”的实验，发现三类特征对于可信依存弧识别问题而言，都有所贡献。其中，文本相关的特征贡献最小，而句法相关的特征贡献最大。最后，我们进行了两种置信度分析的方法的实验对比，实验结果表明：从三种评价方式来看，基于Logistic回归的方法都要优于近似边缘概率的方法。同时，从效率上来考量，基于Logistic回归的方法也有较大的优势。

第 5 章 依存句法分析置信度的应用

信息抽取、自动问答等上层应用都对高精度的依存句法分析工具提出强烈需求，然而之前的研究发现，由于目前依存句法分析准确率不高以及受到领域迁移的限制，句法分析很难真正对应用产生实质性的帮助作用。因此，在本章中，我们通过置信度分析的研究，将依存句法分析结果的置信度较为准确地计算出来。上层应用根据置信度值，或只使用置信度高的依存分析结果，或将置信度信息作为一种特征融入已有系统，从而提高应用系统的准确率。为此，在本章中，我们首先选择篇章级情感分析作为信息抽取的一种代表性应用，以验证依存句法置信度信息的有效性。

依存句法分析置信度分析的另一个应用是依存句法分析自身。我们知道，在主动学习（Active Learning）以及一些半指导或无指导学习方法（例如：自学习）中，都利用了自动预测结果的置信度信息。在主动学习的过程中，需要迭代地从未标注数据中优先选择最具训练效用（测不准）的样本，以减少那些对提高分类器性能帮助不大的冗余样本的人工标注。而对样本训练效用的估计，往往是通过其置信度来进行衡量。在自学习过程中，首先对大规模无依存结果的语料库进行依存句法分析，并将置信度高的分析结果加入原始训练语料库中，从而增大训练语料的规模，提高依存句法分析的准确率。自学习方法对提高依存句法分析、尤其是跨领域的分析具有一定的帮助。前人对依存句法分析树的置信度研究也多应用于此。然而，在与上层应用中遇到的问题一样，这种基于整棵句法依存树的置信度研究存在着精确率（Precision）与召回率（Recall）不高的问题，因此，本章尝试将依存弧的置信度信息应用于一种半指导的依存句法分析方法，以提高其性能。

同时，为分别验证本文在基于转移的依存分析模型与基于图的依存分析模型之上提出的置信度估计方法的有效性，在篇章级情感分析的应用中，我们使用的是基于图的依存分析模型以及相应的最佳置信度计算方法——KD-Fix（基于固定高斯分布的重采样方法）；在半指导依存分析任务中，则使用基于转移的依存分析模型及其最佳置信度估计方法——基于Logistic回归的方法。

5.1 篇章级情感分析

随着Web2.0的发展，互联网上产生了大量由用户直接参与的，对于诸如事

件、人物、商品、服务等评论信息。这些评论信息在某种程度上表达了人们的情感倾向，如“喜”，“厌”，“批评”，“称赞”等。这些评论信息是非常有价值的，潜在用户可以通过浏览这些评论信息，来了解人们对于某产品的看法。而随着评论信息的极速膨胀，仅仅依靠人工的方法难以应对海量信息的收集与处理。因此，情感分析应运而生。

按照分析的粒度不同，情感分析可分为篇章级、句子级、短语级等若干个层次。本文则以篇章级情感分析作为应用场景，以检验依存句法置信度信息的有效性。

篇章级情感分析的基本任务是是对于一条评论（关于酒店、宾馆、电影、商品等），判断其为褒义（positive）或贬义（negative），可视为一个文本分类问题。B.Pang的研究工作表明，使用支持向量机，辅以所有词的一元特征（Unigram），可以达到最好的性能[57]。因此，我们将这种方法作为Baseline方法，以进行比较。

然而，在很多时候，词的一元特征并不具备足够强的判别能力。在某些情形，词的搭配更为重要，尤其是远距离搭配。例如，对于“高”这个词，在很多情况下，表达一种褒义的评价，然而在有些情形，却表达贬义的情感。

图 5-1给出了“高”字分别表达褒义和贬义的情形。可见，只根据词的一元特征所得到的判别模型的能力是有限的，因此，我们需要引入更多的特征，例如词语搭配。依存弧能够很好地表达图中的搭配信息。由此，我们增加了所有依存词搭配，作为情感分类的特征，以此构建第二个情感分类系统作为对比。

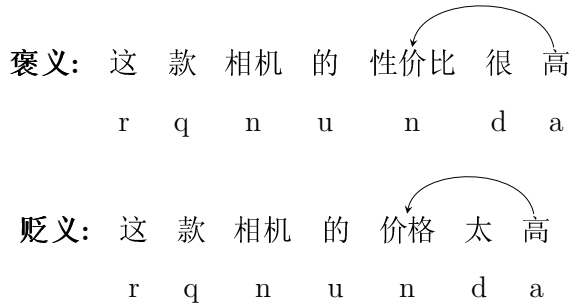

图 5-1 “高”分别表达褒义和贬义的情形

最后，我们利用依存弧的置信度信息，对添加到情感分类系统的依存搭配特征进行过滤，从而构建最终的情感分类系统。置信度的阈值设置为0.7，意味着使用较高置信度的依存词搭配特征。

我们使用 *NLPIR*[1]提供的中文酒店评论平衡语料“ChnSentiCorp-Htl”作为本

[1]http://www.nlpir.org/

实验的数据集集。该语料一共包含2,000篇关于酒店的评论，其中褒义与贬义的评论各1,000篇。

首先需要对数据进行预处理。我们使用哈工大社会计算与信息检索研究中心开发的语言技术平台LTP（Language Technology Platform）[58]对情感数据进行分词以及词性标注。接着使用基于图的依存分析模型对预处理之后的数据进行依存句法分析，并采用基于Logistic回归的方法估计依存分析结果的置信度。三个系统的构建过程如图5-2所示：

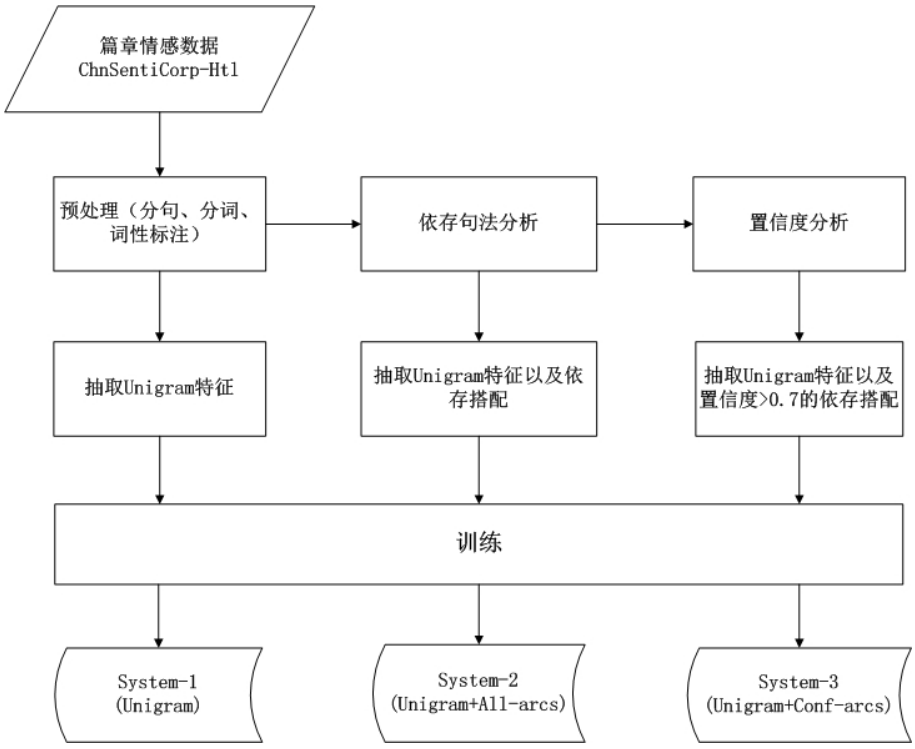

图 5-2 三个情感分类系统的构建过程

本实验中，我们使用svm_light[59] [2]对每篇评论进行分类。表 5-1为“ChnSentiCorp-Htl”语料上的3折交叉验证的平均准确率。

表 5-1 ChnSentiCorp-Htl数据上3-fold准确率

特征	#{依存词搭配}	准确率
Unigrams	0	88.52%
+all-arcs	127,270	89.17%
+conf-arcs	99,293	**89.57%**

从表5-1中可以看出，增加所有依存词搭配特征时，系统准确率由88.52%提

[2]http://svmlight.joachims.org

升至89.17%。若只使用置信度高于0.7的依存词搭配（依存弧），系统性能再度提升至89.57%。同时，根据依存弧置信度对依存词搭配进行过滤之后，所用句法特征数为原来的78%。这意味着，通过对依存弧的置信度分析，能够在一定程度上（并不绝对）进行特征选择，丢掉那些错误的，即置信度较低的依存搭配，从而提高篇章级情感分析的性能。

5.2 半指导依存句法分析

对于依存句法分析而言，标注大规模依存树库的代价是相当昂贵的，而未标注数据则很容易获取。因此，人们越来越多地研究如何利用未标注数据来改善现有的有指导模型。Wenliang Chen提出一种利用大规模自动标注的依存句法分析子结构（包括依存弧）来帮助提高依存句法分析的性能的方法[36]。实验证明，通过加入自动标注结果中的统计信息，可以提高依存句法分析器的性能。然而，自动标注结果中错误的依存子结构对这种半指导依存分析方法是否会有影响呢？本节针对这一问题，在Wenliang Chen的工作基础之上进行了相应的实验。

5.2.1 半指导依存句法分析系统的构建

首先，我们构建一个半指导依存句法分析系统，构建过程如下：

(1) 训练一个基于转移的依存分析模型，作为基准模型。在基准模型中，使用CDT(Chinese Dependency Treebank)作为训练数据，同时，使用到的特征如表 5-3所示，其中特征中各个符号的意义见表5-2。针对每个状态(State)的三元组形式，我们用以下的形式来表达一个特征：[Map](Address)_Attribute，其中Address为地址信息，可表示栈、输入缓冲中任意位置；Attribute为属性信息，可表示词型、原型、词性、依存关系等；Map为映射函数，可取位于Address地址的单词的父亲、儿子结点，构成复合地址。除了表 5-3中所示的基本特征之外，我们也手动对某些词性相关的特征进行了组合。

(2) 对未标注数据进行预处理，并使用基准模型进行依存分析。我们采用已经分词好的人民日报的数据作为未标注数据，并且使用自动词性。我们对语料进行了简单地筛选，选出长度小于60的句子，一共288,385个句子，进行句法分析。

(3) 将未标注数据的自动依存分析结果中，距离小于4的依存弧及其频率抽取出来，并存储在一个列表中，记为DepList。

表 5-2 特征表示

Address	Map	Attribute
STACK[0,1,...,n]	Head: 父亲节点	FORM：词型
BUFFER[0,1,...,n]	RDEP：最右儿子节点	LEMMA：原型
	LDEP：最左儿子节点	CPOSTAG：粗粒度词性
	RNDEP：左边最近儿子节点	POSTAG：细粒度词性
		DEPREL：依存关系

表 5-3 基准依存分析模型中使用的特征

词型相关	词性相关	依存弧相关
STACK[0]_FORM	STACK[0]_CPOSTAG	LDEP(STACK[0])_DEPREL
BUFFER[0]_FORM	BUFFER[0]_CPOSTAG	LDEP(BUFFER[0])_DEPREL
BUFFER[1]_FORM	BUFFER[1]_CPOSTAG	RDEP(STACK[0])_DEPREL
STACK[0]_LEMMA	STACK[1]_CPOSTAG	RNDEP(STACK[0])_DEPREL
BUFFER[0]_LEMMA	BUFFER[2]_CPOSTAG	RDEP(BUFFER[0])_DEPREL
BUFFER[1]_LEMMA	BUFFER[3]_CPOSTAG	

在抽取的过程中，我们考虑了依存弧的跨度（即距离）以及方向。例如，在句子：“我/即将/毕业/了”中，“我”与“毕业”在其依存句法分析结果中构成一条依存弧，且由“毕业”指向“我”（向左），若该依存关系在所有未标注语料中出现48次，则可在DepList中插入一条记录：(D2-L-48)，其中D2表示“我”与“毕业”的距离，L表示依存弧的方向。由于频率是一个实数值，若直接使用则难免会带来特征稀疏的问题，因此，我们将其映射至4个桶（bucket），分别为**ONE**、**LOW**、**MID**、**HIGH**。其中，**ONE**表示频率为1，即只出现一次；**LOW**表示出现次数在2-7之间；**MID**为出现次数在8-20之间；**HIGH**则表示出现次数大于等于21次的依存关系。在之前的例子中，往DepList插入的记录应为：(D2-L-HIGH)。

(4) 基于第(3)步所得到的DepList，为基准模型增加新的特征。

如前所述，在基于转移的依存分析中，每次执行转移操作，都试图判断当前状态下的栈顶元素（记为TOP）与列表头部元素（记为NEXT）之间是否构成依存弧，且方向如何。因此，我们增加的特征也是基于TOP与NEXT所构成的词对（TOP, NEXT）。在每次对转移动作进行决策时，都在DepList中查找词对（TOP, NEXT）所对应的记录，再联合TOP与NEXT在原句中的距离，构成新的特征。例如，对于句子“我/w_1/w_2/w_3/毕业”，在某状态下，TOP元素为“我”，

NEXT元素为“毕业”，两者距离为4(3+)，对应DepList中的记录为“D1-L-HIGH”，则新特征可表示为：“TN0:3+:D1-L-HIGH”，其中TN0表示词对（TOP, NEXT）。另外，还可以抽取更丰富的上下文相关的特征，例如TOP与NEXT的后继词所构成的词对（TOP, NEXT+1），由TN1表示；以及TOP与NEXT的前一个词所构成的词对（TOP, NEXT-1），由TN_1表示。

图5-3为新特征构造的一个图示。

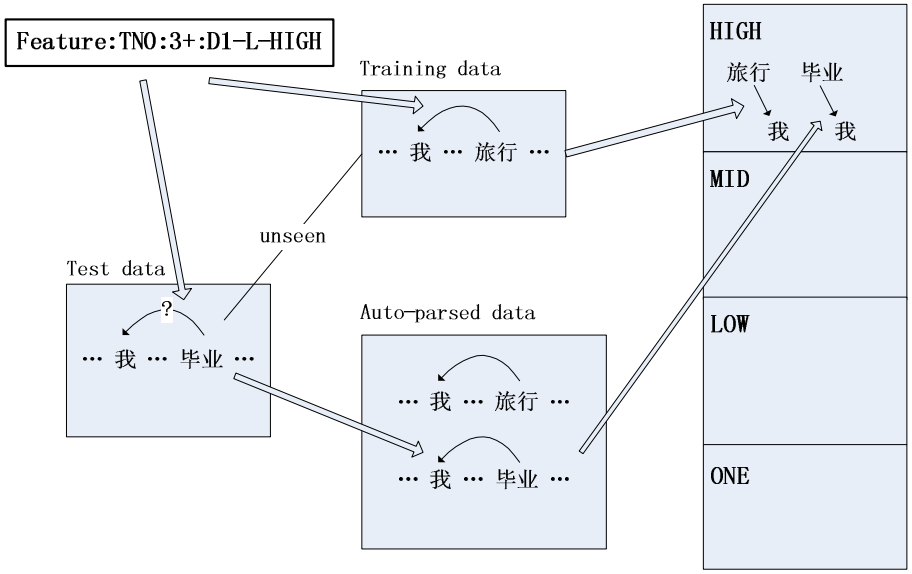

图 5-3 从未标注数据中抽取的新特征

(5) 加入新的特征，训练依存句法模型。

上述过程中存在的一个问题是，从自动句法分析结果中抽取出来的依存关系信息（包括其频率分布）是否真实。显然DepList中的频率统计信息全部是基于自动分析结果，其可信度并不高。因此，我们在上述方法中作了简单的修改：在第(3)步构造DepList时，丢掉置信度低的依存关系，而保留置信度较高的依存关系。其他则保持过程一致。

5.2.2 实验及分析

我们采用CDT数据集训练基准模型，利用未标注的人民日报语料作为未标注数据，并使用基准模型对其进行依存句法分析，从而构建DepList。基于DepList，再在基准模型的基础之上融入新的特征，形成模型2（+Semi-features）。整个过程可视为一种半指导依存句法分析的方法。最后，我们采用基于转移的依存分析中性能最好的置信度估计方法——KD-Fix，对未标注数据的自动句法分析结果进行置信度分析。并在置信度较高的自动分析结果基

础上重新构建DepList，并重复前述过程，得到模型3（+Conf-semi-features）。表5-4为三个模型最终的性能：

表 5-4 半指导依存句法分析性能

Feature	LAS(%)	UAS(%)
Basic-features	76.64	79.76
+Semi-features(D1)	76.95	80.09
+Semi-features(D1,D2)	77.14	80.20
+Semi-features(D1,...,D5)	77.23	80.41
+Conf-semi-features(D1,...,D5)	**77.36**	**80.53**

在模型2（+semi-features）中，本实验分别构造了3个子模型：为了使从未标注数据的自动分析结果中抽取出来的依存搭配信息尽量可靠，即准确率尽量高，我们首先抽取短距离依存弧进行实验（D1，D2）；随后，我们再使用所有距离从1-5的依存搭配特征构建模型，并在该模型基础之上通过置信度信息对特征进行筛选，以矫正其频率分布。

从表5-4中可以得到如下结论：1.通过引入未标注数据中的统计信息作为模型的特征，能够较显著地提升基于转移的依存句法分析模型的性能（LAS提高0.59%）；2.利用自动依存分析结果的置信度信息，对新特征频率分布进行矫正，能够进一步提升未标注数据的效用（0.59% → 0.72%）。

需要注意的是，置信度信息在这里的功能仅仅是对特征频率分布进行矫正，由于——1. 未标注数据也来自人民日报，因此，与训练数据同领域；2. 抽取的依存搭配距离在5以内——这两点原因，导致我们所抽取的依存搭配本身可信度较高，从而导致矫正的效果并非特别明显，因此，最终融入置信度信息所带来的提高相对而言也不甚显著（0.13%）。

5.3 本章小结

本章在两种应用场景下，验证了依存句法分析置信度信息在实际应用中的有效性。一种场景是信息抽取的应用，另一种是依存句法分析自身。

对于信息抽取，我们选择了篇章级情感分析的任务。情感分析是近年来比较火的一个研究课题。随着Web2.0的蓬勃发展，互联网上产生了大量的关于电影、商品、事件的评论信息。这些评论信息中往往带有评论者的个人感情色彩。如何有效地管理和利用这些信息成为一种挑战，而篇章级情感分析的基本任务则是对评论的褒贬性情感倾向进行分类。由于在篇章级情感分类的任务

中，词的Unigram特征并不具有足够强的判别能力，我们从句法依存结果中提取出了依存词搭配，作为Unigram之外的另一种特征。然而，自动句法分析结果中，并非所有依存词搭配都是正确的，那么错误的搭配会不会影响情感分类的性能呢? 针对这个问题，本章设计了三个模型，分别只使用Unigram特征、使用Unigram特征+全部句法依存搭配特征，以及Unigram特征+置信度高的依存搭配特征。实验结果证明，若根据置信度对句法依存搭配特征进行裁剪，那么系统能够在更少的特征之下，取得更优的性能。由此可见，置信度在实际应用中是有效的。

对于依存句法分析自身，我们借鉴Wenliang Chen从大规模未标注数据的自动分析结果中抽取子树的思想，首先在基于转移的依存句法分析模型的基础之上，构建了一套半指导的依存句法分析系统。该方法从自动标注的大量未标注数据中抽取出依存词搭配信息，并将其作为依存分析模型的新特征。我们利用置信度信息对该方法中从未标注数据中采集的信息进行矫正，从而更进一步地提高了依存分析系统的性能，然而由于本实验中使用的未标注数据与训练集同领域，因此特征分布的矫正所带来的性能提升并不是非常显著。

结　论

本文提出了一个新的自然语言处理问题：依存句法分析的置信度估计。该问题的提出主要受上层应用所驱动。在信息抽取、自动问答等许多实际应用问题中，对句法分析的结果往往非常依赖。然而，受到语言自身歧义以及领域迁移等问题的影响，目前的依存分析器在实际应用中的性能一般，短时间内不会有很大的突破。因此，本文不着眼于如何提高依存分析性能，而是从另一个角度出发，通过对依存句法分析结果进行细粒度（依存弧）的置信度估计，来更好地支撑上层应用。本文针对依存分析结果的置信度估计以及应用展开了一系列的研究。首先，对于不同依存句法分析模型，本文分别提出了一套较为准确的置信度估计的方法以及相应的评价准则。同时也从各个角度对每种方法的优劣进行了客观的比较。最后，本文将置信度信息应用于两种特定的应用：篇章级情感分析以及半指导依存句法分析，实验结果表明，对置信度信息的合理利用能够提高上层应用的性能。

具体来讲，本文的创新点以及主要贡献包括以下几个方面：

(1) 提出了依存句法分析置信度估计的新问题。针对目前依存句法分析准确率不高，尤其是受到领域迁移的影响，难以满足实际应用的问题，我们提出了依存句法分析置信度估计的问题。我们分析了之前研究中对整棵句法树的可信度进行识别的缺陷，提出了新的置信度分析对象：依存分析树中的每条依存弧。

(2) 提出了在基于转移的依存句法分析模型上的三类依存弧置信度估计的方法。针对基于转移的依存分析的特点，我们先后提出了基于似然概率、基于重采样以及基于K-Best依存分析结果加权投票等三类方法，计算依存分析树中的每条弧的置信度。其中，基于似然概率的方法最为直观，但是仅限于条件概率的分类模型，比如最大熵。根据先验分布的参数设定方式的不同，基于重采样的的估计方法分为两种：KD-Fix与KD-PC。KD-Fix认为模型中任一特征的权值服从同一高斯分布，而KD-PC则将这种先验分布融入模型的学习过程，利用Confidence-Weighted学习机制得到每一维权值的期望以及方差，并在此基础之上进行重采样。基于K-Best依存分析结果的方法以加权投票的方式进行置信度计算。我们采用期望准确率-实际准确率曲线以及均方根误差两种评价准则对这四种方法进行了对比，实验结果表明，基于KD-Fix的方法的置信度的估计最为准确。

(3) 提出了在基于图的依存句法分析模型上的两类依存弧置信度估计的方法。针对基于图的依存分析的特点，我们先后提出了基于近似边缘概率的方法以及基于Logistic回归的方法。在基于近似边缘概率的方法中，我们对基于图的线性模型作了对数-线性假设，在此基础之上估计依存弧的边缘概率，并将此边缘概率作为该依存弧的置信度。基于Logistic回归的方法是一种有指导的置信度分析方法，以分类的视角看待置信度的估计问题，即将依存弧分为两类，可信与不可信。与一般分类不同之处在于：1.在分类的同时，需要给出依存弧属于某一类别的概率（置信度），因此，我们选择一种概率模型──Logistic回归作为该问题的分类器；2.类样本分布不平衡。由此带来的问题是，如何在模型学习的过程中克服这种不平衡性，以及如何有效合理地评价分类器的性能。针对这些问题，本文分别进行了分析，并给出了相应的解决办法。最后，我们采用了PR曲线、ROC曲线、P@N曲线以及曲线下面积AUC-PR和AUC-ROC作为评价准则，在这三种准则下，Logistic回归的方法优于近似边缘概率的方法。同时，从效率上来考虑，Logistic回归也有较大优势。

(4) 从应用的角度证明了依存句法分析置信度信息的有效性。本文设定了两个应用场景，一是篇章级情感分析（一种信息抽取）、另外一个是半指导依存句法分析。前者着眼于上层应用，后者则针对依存句法分析自身。实验表明，通过合理地利用置信度信息，应用系统的性能都得到了一定程度的提升。

尽管本文在目前已经取得了一定的阶段性研究成果，但是，仍然存在许多需要改进的地方，也存在一些具有挑战性的问题需要进一步研究，主要包括以下几部分：

(1) 置信度分析对象的泛化。本文所设定的置信度分析对象是单条依存弧。实际上，在很多应用中，对依存句法树中的句法路径或是子树的置信度也有所需求。例如实体关系抽取。未来可考虑探索更复杂依存句法子结构的置信度分析，为上层应用提供更全面的支撑。

(2) 置信度分析方法的改进。在本文在基于图的依存模型中所提出的基于Logistic回归的置信度估计方法中，隐含了一个基本假设，即每条依存弧的置信度是相互独立的。因此，在对单条依存弧进行置信度估计时，并未考虑其相邻依存弧或是子孙/父亲节点的依存弧的置信度信息。而这种假设尽管简单，却有不合理之处。未来可考虑在建模过程中，加入依存弧之间的依赖。

(3) 置信度分析的应用。由于诸多应用对依存句法分析都提出了需求，因此，未来期望能够将置信度信息应用于更多的实际应用中，充分发挥其作用。其中，跨领域的依存句法分析将是一个非常好的应用场景。

参考文献

[1] Kübler S, McDonald R, Nivre J. Dependency parsing[J]. Synthesis Lectures on Human Language Technologies, 2009, 1(1):1–127.

[2] Buchholz S, Marsi E. CoNLL-X shared task on multilingual dependency parsing[C]//Proceedings of the Tenth Conference on Computational Natural Language Learning. .[S.l.]: [s.n.] , 2006:149–164.

[3] Nilsson J, Riedel S, Yuret D. The CoNLL 2007 shared task on dependency parsing[C]//Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL. .[S.l.]: [s.n.] , 2007:915–932.

[4] Surdeanu M, Johansson R, Meyers A, et al. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies[C]//Proceedings of the Twelfth Conference on Computational Natural Language Learning. .[S.l.]: [s.n.] , 2008:159–177.

[5] 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: Shared Task. .[S.l.]: [s.n.] , 2009:1–18.

[6] Chang C, Kayed M, Girgis R, et al. A survey of web information extraction systems[J]. Knowledge and Data Engineering, IEEE Transactions on, 2006, 18(10):1411–1428.

[7] Pasca M. Open-domain question answering from large text collections[J]. Computational Linguistics, 2003, 29(4):665–667.

[8] Cui H, Sun R, Li K, et al. Question answering passage retrieval using dependency relations[C]//Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. .[S.l.]: [s.n.] , 2005:400–407.

[9] Fundel K, Küffner R, Zimmer R. RelEx—Relation extraction using dependency parse trees[J]. Bioinformatics, 2007, 23(3):365–371.

[10] McClosky D, Charniak E, Johnson M. Automatic domain adaptation for parsing[C]//Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. .[S.l.]: [s.n.] , 2010:28–36.

[11] Zhang M, Zhang J, Su J, et al. A composite kernel to extract relations between entities with both flat and structured features[C]//Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. .[S.l.]: [s.n.] , 2006:825–832.

[12] Hays D, of Scientific Research U S A F O, Corporation R. Studies in Machine Translation–10: Russian Sentence-structure Determination[M].[S.l.]: Rand, 1960.

[13] Hays D. Dependency theory: A formalism and some observations[J]. Language, 1964, 40(4):511–525.

[14] Hays D. Readings in automatic language processing[M].[S.l.]: Elsevier, 1966.

[15] Fraser N. Dependency parsing[D].[S.l.]: University College, London, 1993.

[16] Fraser N. Dependency grammar[J]. Encyclopedia of Language and Linguistics, 1994:860–864.

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

[18] Nivre J. Algorithms for deterministic incremental dependency parsing[J]. Computational Linguistics, 2008, 34(4):513–553.

[19] Nivre J, Hall J, Nilsson J, et al. MaltParser: A language-independent system for data-driven dependency parsing[J]. Natural Language Engineering, 2007, 13(02):95–135.

[20] Koo T, Collins M. Efficient third-order dependency parsers[C]//Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. .[S.l.]: [s.n.] , 2010:1–11.

[21] Sagae K, Lavie A. Parser combination by reparsing[C]//Proceedings of the human language technology conference of the NAACL, companion volume: short papers. .[S.l.]: [s.n.] , 2006:129–132.

[22] Nivre J, McDonald R. Integrating graph-based and transition-based dependency parsers[J]. Proceedings of ACL-08: HLT, 2008:950–958.

[23] Zhang Y, Clark S. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. .[S.l.]: [s.n.] , 2008:562–571.

[24] Koo T, Carreras X, Collins M. Simple semi-supervised dependency parsing[J]. 2008.

[25] Chen W, Kazama J, Uchimoto K, et al. Improving dependency parsing with subtrees from auto-parsed data[C]//Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2. .[S.l.]: [s.n.] , 2009:570–579.

[26] 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. .[S.l.]: [s.n.] , 2009:1222–1231.

[27] McClosky D, Charniak E, Johnson M. Effective self-training for parsing[C]//Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. .[S.l.]: [s.n.] , 2006:152–159.

[28] Bansal M, Klein D. Web-scale features for full-scale parsing[C]//Proceedings of ACL. .[S.l.]: [s.n.] , 2011:693–702.

[29] Zhou G, Zhao J, Liu K, et al. Exploiting web-derived selectional preference to improve statistical dependency parsing[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. .[S.l.]: [s.n.] , 2011:1556–1565.

[30] Yates A, Schoenmackers S, Etzioni O. Detecting parser errors using web-based semantic filters[C]//Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. .[S.l.]: [s.n.] , 2006:27–34.

[31] Reichart R, Rappoport A. An ensemble method for selection of high quality parses[C]//ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. .[S.l.]: [s.n.] , 2007，45:408.

[32] Kawahara D, Uchimoto K. Learning reliability of parses for domain adaptation of dependency parsing[J]. IJCNLP’08, 2008.

[33] Ravi S, Knight K, Soricut R. Automatic prediction of parser accuracy[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. .[S.l.]: [s.n.] , 2008:887–896.

[34] Settles B. Active learning literature survey[J]. University of Wisconsin, Madison, 2010.

[35] Van Noord G. Using self-trained bilexical preferences to improve disambiguation accuracy[C]//Proceedings of the 10th International Conference on Parsing Technologies. .[S.l.]: [s.n.] , 2007:1–10.

[36] Chen W, Kawahara D, Uchimoto K, et al. Dependency parsing with short dependency relations in unlabeled data[C]//Proceedings of IJCNLP. .[S.l.]: [s.n.] , 2008.

[37] Nivre J. An efficient algorithm for projective dependency parsing[C]//Proceedings of the 8th International Workshop on Parsing Technologies (IWPT. .[S.l.]: [s.n.] , 2003.

[38] Collins M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms[C]//Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. .[S.l.]: [s.n.] , 2002:1–8.

[39] 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. .[S.l.]: [s.n.] , 2005:91–98.

[40] Eisner J. Three new probabilistic models for dependency parsing: An exploration[C]//Proceedings of the 16th conference on Computational linguistics-Volume 1. .[S.l.]: [s.n.] , 1996:340–345.

[41] Carreras X. Experiments with a higher-order projective dependency parser[C]//Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL. .[S.l.]: [s.n.] , 2007，7:957–961.

[42] 李正华, 车万翔, 刘挺. 基于柱状搜索的高阶依存句法分析[J]. 中国计算机语言学研究前沿进展(2007-2009), 2009.

[43] Crammer K, Dekel O, Keshet J, et al. Online passive-aggressive algorithms[J]. The Journal of Machine Learning Research, 2006, 7:551–585.

[44] Crammer K, Singer Y. Ultraconservative online algorithms for multiclass problems[J]. The Journal of Machine Learning Research, 2003, 3:951–991.

[45] McDonald R, Nivre J. Characterizing the errors of data-driven dependency parsing models[C]//Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). .[S.l.]: [s.n.] , 2007:122–131.

[46] 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. .[S.l.]: [s.n.] , 2011:1385–1394.

[47] Covington M. A fundamental algorithm for dependency parsing[C]//Proceedings of the 39th annual ACM southeast conference. .[S.l.]: [s.n.] , 2001:95–102.

[48] Gildea D, Jurafsky D. Automatic labeling of semantic roles[J]. Computational Linguistics, 2002, 28(3):245–288.

[49] Sag I, Baldwin T, Bond F, et al. Multiword expressions: A pain in the neck for NLP[J]. Computational Linguistics and Intelligent Text Processing, 2002:189–206.

[50] Mejer A, Crammer K. Confidence in structured-prediction using confidence-weighted models[C]//Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. .[S.l.]: [s.n.] , 2010:971–981.

[51] Dredze M, Crammer K, Pereira F. Confidence-weighted linear classification[C]//Proceedings of the 25th international conference on Machine learning. .[S.l.]: [s.n.] , 2008:264–271.

[52] Liu T, Ma J, Li S. Building a dependency treebank for improving Chinese parser[J]. Journal of Chinese Language and Computing, 2006, 16(4):207–224.

[53] Koo T, Globerson A, Carreras X, et al. Structured prediction models via the matrix-tree theorem[C]//Proc. EMNLP. .[S.l.]: [s.n.] , 2007.

[54] Bishop C, en ligne) S S. Pattern recognition and machine learning[M]. Vol. 4.[S.l.]: springer New York, 2006.

[55] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves[C]//Proceedings of the 23rd international conference on Machine learning. .[S.l.]: [s.n.] , 2006:233–240.

[56] Bohnet B. Very high accuracy and fast dependency parsing is not a contradiction[C]//Proceedings of the 23rd International Conference on Computational Linguistics. .[S.l.]: [s.n.] , 2010:89–97.

[57] Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques[C]//Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. .[S.l.]: [s.n.] , 2002:79–86.

[58] Che W, Li Z, Liu T. Ltp: A chinese language technology platform[C]//Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations. .[S.l.]: [s.n.] , 2010:13–16.

[59] Joachims T. Making large scale SVM learning practical[J]. 1999.

攻读硕士学位期间发表的论文及其他成果

（一）发表的学术论文

[1] 郭江，车万翔，刘挺. 汉语语义依存分析. 智能计算机与应用, 2011, 1(2): 58-62.

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《依存句法分析的置信度研究》，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

值此论文完成之际，谨向所有关心、帮助过我的老师、同学、亲友表示衷心的感谢。

感谢我的导师车万翔老师，在我硕士的两年里，对我在学业以及科研上认真的指导。两年中，做过很多事情，作为一个刚摸索在科研路上的小兵，难免磕磕绊绊，常有挫折。每一次面对被拒的论文、面对糟糕的实验结果，心情沮丧时，车老师总会笑着告诫我：这就是科研，要容忍失败。我想，不仅仅是在对科研的认知，还有对待人生，对待成长的心态，我都从车老师身上受益匪浅。漫漫读博路，不知又会遇到多少坎坷与磨难，我会一直保持着如此乐观豁达的心态对待科研，对待生活。感谢车老师。

感谢刘挺教授为我提供如此融洽和积极向上的实验环境，使我每日清晨踏进实验室时，都能感受到一股催人奋进的力量。深深地感激刘老师为实验室的我们所做的一切。也感谢美丽的秦老师和和蔼的张老师给我许多的帮助和指导。美丽的赛尔家园正散发着蓬勃的生机，相信在大家的一起努力之下，实验室的明天定会更加绚丽。

感谢所有聪明美丽淳朴无邪的赛尔人，你们是光，你们是电，你们是美丽的神话。我们一路前行，正因为有你们的陪伴，路途才显得不那么遥远和孤独。两年里，横扫学士楼的超级无敌大饭团、羽毛球场上的3V3、赛尔家园的鲜花与鸡蛋、地下组织里的扯淡与调侃，都是这一生难以忘却的回忆与温暖。

感谢LA组的正华师兄、梅山师兄、还有已经毕业的大华、鑫哥，静姐，兄弟唐国华、陆子龙、邓知龙、丁宇、一佳。以你们为镜，我看到了自己身上许多的不足。正华师兄实事求是不浮夸的科研态度让人倾慕、梅山师兄认真深刻时而有点小变态的工作作风也让人倍感亲切。子龙看似上下文无关的跳跃式思维实际上饱含着深刻的逻辑、知龙朴实典雅让人毫无距离感的笑容让人心生亲近；丁宇是个好姑娘，一佳是个乖孩子。你们前途无限！

感谢10级所有的兄弟姐妹，一切尽在不言中，但我还是试图用四个字来表达我对你们的倾慕：罗磊温润如玉、炜鹏诙谐善思、汉东谦谦君子、彪叔癫狂不羁、糖果清新无邪、钰帝贪玩重义、小谢细腻重口、阿燊慧之思念、一搏文艺性情。感谢你们的出现。

最后要感谢我的父母，谢谢你们对我无条件的支持与无私的关爱，至今已二十余春秋。此情难寄，此恩难谢。