基于双向依赖树特征的方面术语提取改进

Improving Aspect Term Extraction with Bidirectional Dependency Tree Representation.
Huaishao Luo, Tianrui Li, Bing Liu, Bin Wang and Herwig Unger. TASLP, 2019, 27(7):1201-1212.

摘要

方面术语提取是基于方面的情感分析的重要子任务之一。 以前的研究表明，使用依赖树结构表示对于这项任务是有希望的。 然而，大多数依赖树结构只涉及依赖树上的一个方向传播。 本文首先提出了一种新的双向依赖树网络，从给定的句子中提取依赖结构特征。 关键的思想是显式地将从自下而上和自上而下的传播中获得的两种表示合并到给定的依赖句法树上。 然后我们开发了一个端到端框架来集成嵌入式表示，然后用BiLSTM+CRF以学习树状结构和顺序特征，从而解决方向术语提取问题。 实验结果表明，该模型在四个基准SemEval数据集上优于最先进的基本模型。

1 介绍

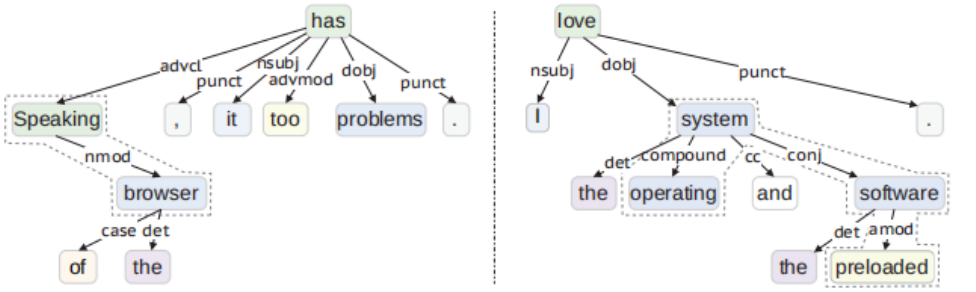
方面术语提取(ATE)是提取人们表达的意见实体的属性（或方面）的任务。 它是基于方面的情感分析中最重要的子任务之一。 如表1所示，前两句中的“设计”、“气氛”、“工作人员”、“酒吧”、“饮料”和“菜单”是餐厅评论的方面术语，最后两句中的“操作系统”、“预装软件”、“硬盘”、“Windows”和“驱动程序”是笔记本电脑评论的方面。

> 表 1： 用户评论的示例，以粗体标记方面术语。

No.	评论
1	**设计**和**氛围**都一样好。
2	**工作人员**非常善良，训练有素，他们很快，他们总是迅速地跳到**酒吧**后面，准备**饮料**，他们知道**菜单**上每一项的细节，并提出极好的推荐。
3	我喜欢**操作系统**和**预装软件**。
4	**硬盘**似乎也有问题，因为某些时候**windows**加载，但声称找不到任何**驱动程序**或文件。

目前存在的ATE方法可以分为有监督和无监督两种。无监督方法主要基于主题建模、句法规则和终身学习。有监督方法主要基于有条件随机场(CRF)。

本文将基于CRF的模型作为序列标记任务。在以前的基于通用报告格式的ATE模型中使用了三种主要类型的特征。第一类是传统的自然语言特征，例如，句法结构和词汇特征。第二种类型是基于跨域知识的特性，这是有用的，因为跨域有许多共享方面，尽管每个实体/产品是不同的。最后一种类型是通过深度学习模型学习的深度学习特征，近年来已被证明对ATE非常有用。



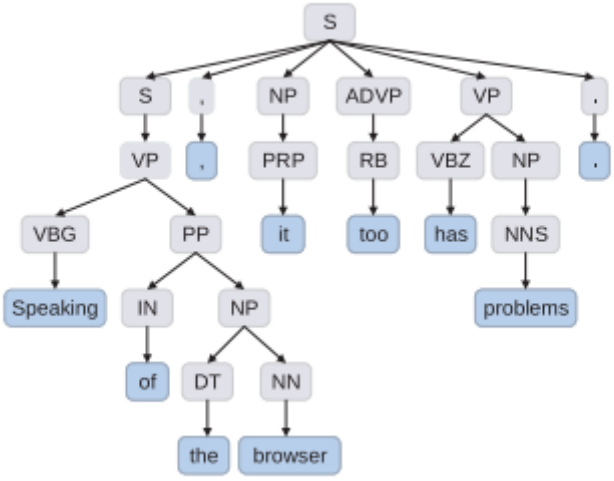
a) Speaking of the browser, it too has problems. b) I love the operating system and the preloaded software.

> 图1：依赖关系示例(由Stanford CoreNLP3.8.0的基本依赖项生成)。 每个节点是一个单词，每条边是两个单词之间的依赖关系。

深度学习特征一般包括顺序表示和树结构表示特征。

顺序表示是指句子的语序。结构表征特征来源于句子的句法结构，它代表了词与词之间的内在逻辑关系。图1显示了依赖结构的两个例子，其中每个节点是句子的一个单词，每个边是单词之间的依赖关系。例如，$Speaking \xrightarrow{nmod} browser$ 意味着*speaking*是*browser*的名义修饰符。 这种关系在ATE中是有用的。例如，给定*System*作为一个方面术语，*Software*可以通过 $system \xrightarrow{conj} software$ 关系提取为一个方面术语。因为*conj*意味着*System*和*Software*是通过协调连接（例如，*and*）。 然而，在以前的工作中，树结构表示只考虑了在具有共享权重的解析树上训练的单个传播方向(自下而上的传播。我们进一步利用树结构表示的能力通过考虑到自上而下的传播，这意味着给定*Software*作为一个方面术语，*System*可以通过关系 $software \xrightarrow{conj^{-1}} system$ 提取为一个方面术语，*conj-1*是为了区分不同传播方向而进行的*conj*的反关系。与顺序表示相比，树结构表示能够获得单词之间的长程依赖关系，特别是对于长句，如表1中的第二和第四次评论。

在本文中，我们首先使用来自双向LSTM(BiLSTM)的双向门控制机制来增强树状结构表示)，然后融合树结构和顺序信息来执行方面术语提取。通过将这两个步骤合二为一，我们提出了一个框架"双向依赖树条件随机字段的框架"bidirectional dependency tree conditional random fields (BiDTreeCRF)。具体而言，BiDTree CRF是一个增量框架，由三个主要组成部分组成。第一个组件是双向依赖树网络(BiDTree)，它是递归神经网络的扩展。 它的目标是从给定句子的依赖树中提取树结构表示。 第二个组件是BiLSTM，其输入是BiDTree的输出。树结构和顺序信息在这一层中融合。 最后一个组件是CRF，用于生成标签。 据我们所知，这是第一个融合树结构和顺序信息来解决ATE的工作。 这一新模型导致ATE在现有基线模型上的重大改进。



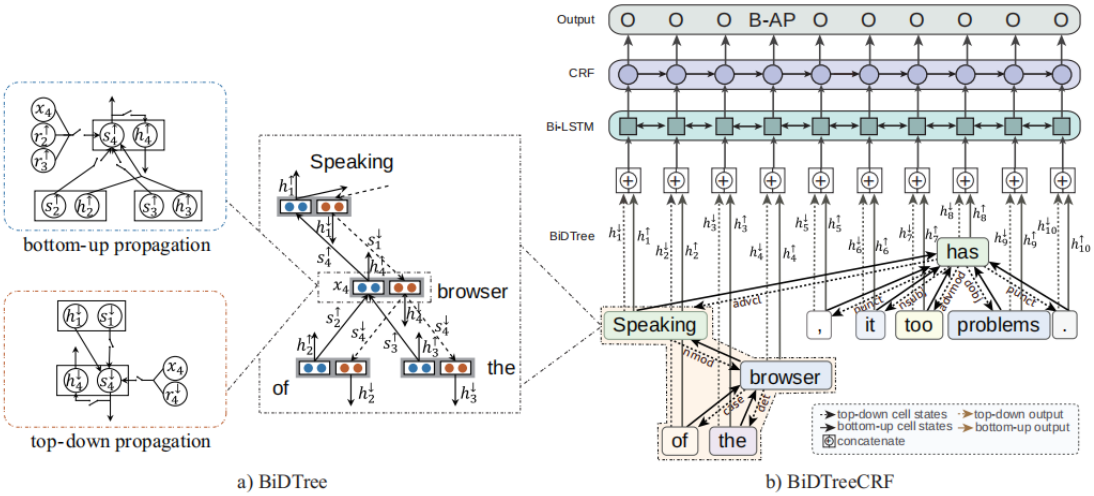
> 图2：一个选区树的示例(由StanfordCoreNLP 3.8.0的选区解析生成)。 蓝色背景的每个节点都是句子中的一个真实单词：*Speaking of the browser, it too has problems.*

我们提出的BiDTree是基于依赖树构造的。与基于选区树的许多其他方法相比（图2）。BiDTree更直接地关注单词之间的依赖关系，因为依赖树中的所有节点都是输入单词本身，但选区树侧重于已识别的短语及其递归结构。

本文的两个主要贡献如下。

- 提出了一种新的双向递归神经网络BiDTree，通过在依赖树上构建双向传播机制来增强树结构表示。 因此，BiDTree可以捕获更有效的树结构表示特征，并获得更好的性能。
- 提出了一种既包含句法信息又包含顺序信息的增量框架BiDTree CRF。 这些信息被输入到CRF层中进行方面术语提取。 集成模型可以以端到端的方式进行有效的训练。

2 模型描述



> 图3：BiDTree和BiDTreeCRF体系结构的说明。 左：BiDTree体系结构，包括自下而上的传播和自上而下的传播；r表示句法关系(例如nmod、case和det)；x是单词；s和h分别表示单元内存和隐藏状态。 右：BiDTreeCRF有三个模块：BiDTree、BiLSTM和CRF

我们所提出的框架的体系结构如图3所示。 它的示例输入是图1中所示的依赖关系。像在第1节中描述中的那样，BiDTree CRF由三个模块（或组件）组成：BiDTree、BiLSTM和CRF。 这些单元将在第2.2和2.3节中详细说明。

2.1 问题陈述

我们从一个特定的域中得到了一个评论句子，由*S={w1，w2，...，wi，...，wN}*表示，其中N是句子长度。 对于任何单词wi∈S，ATE的任务是找到与之对应的标签ti∈T，其中*T={B-AP，I-AP，O}*。 “BAP”、“I-AP”和“O”分别代表一个方面术语的开头，在一个方面术语的内部，以及其他单词。 例如，“*The*/O *picture*/B-AP *quality*/I-AP *is*/O *very*/O *good*/O ./O”是一个带有标签（或标签）的句子，其中方面术语是图片质量。 这种BIO编码方案在NLP任务中得到了广泛的应用，这种任务通常使用基于CRF的方法来解决。

2.2 双向依赖树网络

由于BiDTree是建立在依赖树上的，所以应该首先将句子转换为基于依赖的解析树。 如图1的左边部分所示，依赖树中的每个节点表示一个单词并连接到至少一个其他节点/单词。 每个节点都有一个并且只有一个头字，例如，Speaking是Browser的头，has是Speaking的头，has的头字是ROOT1。 每个节点与其首词之间的边缘是一个句法依赖关系，例如Browser和Speaking之间的nmod用于名词或从句谓语的名词修饰。 图3中的句法关系显示为虚线黑线。生成依赖树后，每个单词wi将用特征向量 $x_{w_i} \in \mathbb{R}^d$ 初始化，它对应于预先训练的单词嵌入 $E \in \mathbb{R}^{d \times |V|}$ 的一列，其中d是单词向量的维数，|V|是词汇

表的大小。如上所述，依赖树的每个关系都从一个头词开始，并指向它的依赖词。 这可以表述如下：调速器节点p及其相关节点 $c_1, c_2, \ldots, c_{n_i} \ldots, c_{n_p}$ 由 $r_{pc_1}, r_{pc_2}, \ldots, r_{pc_i}, \ldots, r_{pc_{n_p}}$ 连接，其中NP是属于p的依赖节点数，$r_{pc_i} \in \mathbb{L}$ 其中L是一组句法关系如*nmod*，*case*，*det*，*nsubj*等。句法关系信息不仅作为网络中编码的特征，而且作为训练权重选择的指南。BiDTree在两个方向使用LSTM工作：自下而上的LSTM和自上而下的LSTM。 自下而上的LSTM显示为实心黑色箭头，自上而下的LSTM显示为图3下部的虚线黑色箭头。应该注意的是，它们不仅在方向上不同，而且在调速器节点和依赖节点上也不同。具体而言，自顶向下的LSTM的每个节点只拥有一个依赖节点，但自下而上的LSTM通常拥有多个依赖节点。 如公式（1）所示。我们将自下而上的LSTM的输出h↑wi和自上而下的LSTM的输出h↓wi连接到hwi中，作为BiDTree对词汇 wi的输出

$$h_{w_i} = [h^{\uparrow}_{w_i}; h^{\downarrow}_{w_i}]. \tag{1}$$

这允许BiDTree捕获全局句法上下文。

设C(P)={c1，c2，...，cni...，cnp}，这是上面描述的节点p的依赖节点集。 在这些符号指令下，BiDTree的自下而上的LSTM首先编码调速器词和相关的句法关系：

$$\mathcal{T}_i = W^{\uparrow(i)} x_{w_p} + \sum_{k \in C(p)} W^{\uparrow(i)}_{r^{\uparrow}(k)} r^{\uparrow}_k, \tag{2}$$

$$\mathcal{T}_o = W^{\uparrow(o)} x_{w_p} + \sum_{k \in C(p)} W^{\uparrow(o)}_{r^{\uparrow}(k)} r^{\uparrow}_k, \tag{3}$$

$$\mathcal{T}_{fk} = W^{\uparrow(f)} x_{w_p} + W^{\uparrow(f)}_{r^{\uparrow}(k)} r^{\uparrow}_k, \tag{4}$$

$$\mathcal{T}_u = W^{\uparrow(u)} x_{w_p} + \sum_{k \in C(p)} W^{\uparrow(u)}_{r^{\uparrow}(k)} r^{\uparrow}_k. \tag{5}$$

然后，BiDTree的自下而上LSTM过渡方程如下：

$$i_p = \sigma\left(\mathcal{T}_i + \sum_{k \in C(p)} U^{\uparrow(i)}_{r^{\uparrow}(k)} h^{\uparrow}_k + b^{\uparrow(i)}\right), \tag{6}$$

$$o_p = \sigma\left(\mathcal{T}_o + \sum_{k \in C(p)} U^{\uparrow(o)}_{r^{\uparrow}(k)} h^{\uparrow}_k + b^{\uparrow(o)}\right), \tag{7}$$

$$f_{pk} = \sigma\left(\mathcal{T}_{fk} + U^{\uparrow(f)}_{r^{\uparrow}(k)} h^{\uparrow}_k + b^{\uparrow(f)}\right), \tag{8}$$

$$u_p = \tanh\left(\mathcal{T}_u + \sum_{k \in C(p)} U^{\uparrow(u)}_{r^{\uparrow}(k)} h^{\uparrow}_k + b^{\uparrow(u)}\right), \tag{9}$$

$$s^{\uparrow}_p = i_p \odot u_p + \sum_{l \in C(p)} f_{pl} \odot s^{\uparrow}_l, \tag{10}$$

$$h^{\uparrow}_p = o_p \odot \tanh(s^{\uparrow}_p), \tag{11}$$

其中IP是输入门，OP是输出门，FPK和FPL是遗忘门，它们是从标准LSTM扩展的。 ↑p和s↑l是存储单元状态，h↑p和h↑k是隐藏状态，σ表示逻辑函数，表示元素乘法，W↑（∗），W（∗）r(K)，U（∗）r(K)是权重矩阵，b（∗）是偏置向量。而r↑(K)是一个映射函数，它将句法关系类型映射到相应的参数矩阵。 ∗∈{i，o，f，u}。 特别是，句法关系rk↑像词向量xwp一样被编码到网络中，但随机初始化。 在我们的实验中，rk↑的大小与xwp相同。

自顶向下的LSTM与自底向上的LSTM具有相同的过渡方程，除了方向和依赖节点的数目。 特别是，自上而下的LSTM的句法关系类型与自下而上的LSTM相反，我们通过添加前缀"l-"来区分它们，例如将*l-nmod*设置为*nmod*。 导致r↓(k)和参数矩阵的差异。 本文将BiDTree的所有权值和偏置向量分别设置为d×d维和d维。 因此，输出hwi是一个二维向量。

作为一个实例，我们给出了图3a)中自下而上传播的具体公式，用于计算单词Browser的输出。在自下而上的方向上，*"of"*和*"the"*两个词分别与目标词*"browser"*相关，分别是"case"和"*det*"。 因此，x4是xbrowser。r↑2和r↑3分别是rcase和rdet。 同样，s↑和h↑的下标2、3和4被替换为相应的单词"of"、"the"和"browser"，以促进理解。 因此，"browser"在自下而上方向上的输出计算如下：

$$
\begin{aligned}
\mathcal{T}_i &= W^{\uparrow(i)} x_{browser} + W_{case}^{\uparrow(i)} r_{case} + W_{det}^{\uparrow(i)} r_{det}, \\
\mathcal{T}_o &= W^{\uparrow(o)} x_{browser} + W_{case}^{\uparrow(o)} r_{case} + W_{det}^{\uparrow(o)} r_{det}, \\
\mathcal{T}_{f(case)} &= W^{\uparrow(f)} x_{browser} + W_{case}^{\uparrow(f)} r_{case}, \\
\mathcal{T}_{f(det)} &= W^{\uparrow(f)} x_{browser} + W_{det}^{\uparrow(f)} r_{det}, \\
\mathcal{T}_u &= W^{\uparrow(u)} x_{browser} + W_{case}^{\uparrow(u)} r_{case} + W_{det}^{\uparrow(u)} r_{det}, \\
i_p &= \sigma\left(\mathcal{T}_i + U_{case}^{\uparrow(i)} h_{of}^{\uparrow} + U_{det}^{\uparrow(i)} h_{the}^{\uparrow} + b^{\uparrow(i)}\right), \\
o_p &= \sigma\left(\mathcal{T}_o + U_{case}^{\uparrow(o)} h_{of}^{\uparrow} + U_{det}^{\uparrow(o)} h_{the}^{\uparrow} + b^{\uparrow(o)}\right), \\
f_{p(case)} &= \sigma\left(\mathcal{T}_{f(case)} + U_{case}^{\uparrow(f)} h_{of}^{\uparrow} + b^{\uparrow(f)}\right), \\
f_{p(det)} &= \sigma\left(\mathcal{T}_{f(det)} + U_{det}^{\uparrow(f)} h_{the}^{\uparrow} + b^{\uparrow(f)}\right), \\
u_p &= \tanh\left(\mathcal{T}_u + U_{case}^{\uparrow(u)} h_{of}^{\uparrow} + U_{det}^{\uparrow(u)} h_{the}^{\uparrow} + b^{\uparrow(u)}\right), \\
s_{browser}^{\uparrow} &= i_p \odot u_p + f_{p(case)} \odot s_{of}^{\uparrow} + f_{p(det)} \odot s_{the}^{\uparrow}, \\
h_{browser}^{\uparrow} &= o_p \odot \tanh\left(s_{browser}^{\uparrow}\right).
\end{aligned}
\tag{12}
$$

*"browser"*的自上而下传播有相同的公式，但方向不同。 特别的，*"Speaking"*一词与目标词*"browser"*有关，其关系是*"l-nmod"*。 因此，x4是xbrowser，r↓4指的是rl-nmod。

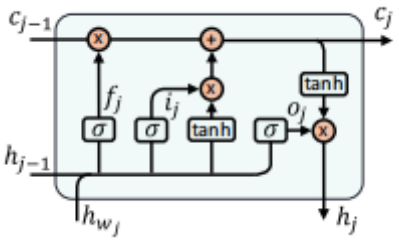
BiDTree的公式与(Miwa和Bansal，2016)中的依赖层相似，主要区别是遗忘门的参数设计。他们的工作定义了具有参数矩阵U↑(F)r↑(K)r(L)2的相关节点的k-遗忘门fpk的参数化。 对应于方程（8）如下：

$$
f_{pk} = \sigma\left(\mathcal{T}_{fk} + \sum_{l \in C(p)} U_{r^{\uparrow}(k) r^{\uparrow}(l)}^{\uparrow(f)} h_k^{\uparrow} + b^{\uparrow(f)}\right).
\tag{13}
$$

在(Tai等人，2015年)中提到，对于大量依赖节点NP，使用额外的参数来灵活地控制信息从依赖到调速器的传播是不切实际的。 考虑到所提出的框架具有可变数量的类型相关节点，我们使用Eq.（8）代替Eq.（13）减少计算费用。 它们的公式和我们的公式的另一个区别是，我们将句法关系编码到我们的网络中，即Eqs.（2-5），这在本文中被证明是有效的。

2.3与双向LSTM的集成

图4： LSTM单元



作为第二个模块，BiLSTM(Graves和Schmidhuber，2005)保持单词之间依赖信息的顺序上下文。如图4所示，LSTM单元位于第j个单词接收BiDTree hwj、前隐藏状态hj-1和前存储单元cj-1的输出，使用以下方程计算新隐藏状hj和新存储单元cj：

$$i_j = \sigma\left(W^{(i)}h_{w_j} + U^{(i)}h_{j-1} + b^{(i)}\right), \quad (14)$$

$$o_j = \sigma\left(W^{(o)}h_{w_j} + U^{(o)}h_{j-1} + b^{(o)}\right), \quad (15)$$

$$f_j = \sigma\left(W^{(f)}h_{w_j} + U^{(f)}h_{j-1} + b^{(f)}\right), \quad (16)$$

$$u_j = \tanh\left(W^{(u)}h_{w_j} + U^{(u)}h_{j-1} + b^{(u)}\right), \quad (17)$$

$$c_j = i_j \odot u_j + f_j \odot c_{j-1}, \quad (18)$$

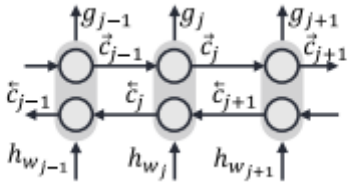
$$h_j = o_j \odot \tanh(c_j), \quad (19)$$

其中ij、oj、fj是具有与BiDTree中对应的含义相同的门，大小为d×2d的W（∗）、大小为d×d的U（∗）是权重矩阵，b（∗）是d维偏置vectors. ∗∈{i，o，f，u}。 我们还将LSTM单元产生的隐藏状态串联在两个方向上，它们属于与输出向量相同的单词，其表示如下：

$$g_j = \left[\overrightarrow{h_j}; \overleftarrow{h_j}\right] \quad (20)$$

BiLSTM的体系结构如图5所示。 此外，每个gj都被一个完整的连接层简化为|T|尺寸，以便在我们的实现中传递给后续的层。

> 图5：双向LSTM



2.4 与CRF的整合

学习的特征实际上是包含树结构和顺序信息的混合特征。所有这些特征都被输入到最后一个CRF层中，以预测每个单词的标签。 本文采用线性链CRF。形式上，让g={g1，g2，...，gj，...，gN}表示BiDTree和BiLSTM层提取的输出特征。通用报告格式的目标是解码标签y={t1、t2、...、Tj、...、tN}的最佳链，其中Tj已在2.1节中描述。 作为一种判别图形模型，CRF受益于考虑邻域内标签/标签之间的相关性，这在序列标记或标记任务中得到了广泛的应用。 设Y(G)表示所有可能的标签，Y0∈Y(G)。计算CRF p(y|g；W，b)的概率如下：

$$p(y|g;W,b) = \frac{\prod_{j=1}^{N}\Psi_j(y_{j-1}, y_j, g)}{\sum_{y'\in Y(g)}\prod_{j=1}^{N}\Psi_j(y'_{j-1}, y'_j, g)}, \quad (21)$$

通常，训练过程使用最大条件似然估计。 对数似然计算如下：

$$L(W,b) = \sum_j \log\, p(y|g;W,b). \quad (22)$$

最后的标记结果以最高的条件概率生成:

$$y^* = \arg\max_{y \in \mathcal{Y}(g)} p(y|g; W, b). \quad (23)$$

这一过程通常由Viterbi算法有效地求解。

2.5 标签结果的解码

一旦产生标记结果，获得给定句子的方面项的最后一步是解码标记序列。 根据T中元素的均值，方便得到方面项。 例如，对于一个句子“W1W2W3W4”，如果标记序列是“B-APB-API-APO”，那么(“W1”，1，2)和(“W2W3”，2，4)是目标方面术语。 对于上面的三重，第一个元素是真正的方面项，第二个元素和最后一个元素分别是句子中的开头（包含)和结尾(独占）索引。 算法1详细给出了这个过程。

Algorithm 1 Decoding from the Labeling Sequence

Input: A labeling sequence $\tau = \{t_1, t_2, \ldots, t_i, \ldots t_N\}$, and its corresponding sentence $S = \{w_1, w_2, \ldots, w_i, \ldots, w_N\}$.

Output: A list of aspect term triples

```
result ← ()
temp ← ""
start ← 0
for i = 1; i ≤ N; i++ do
    if t_i = "O" and temp ≠ "" then
        result ← result + (w_start:i, start, i)
        temp ← ""
        start ← 0
    else
        if t_i = "B-AP" then
            if temp ≠ "" then
                result ← result + (w_start:i, start, i)
            end if
            temp ← t_i
            start ← i
        end if
    end if
end for
if temp ≠ "" then
    result ← result + (w_start:i, start, i)
end if
return result
```

2.6 损失和模型训练

我们等价地使用方程中L(W，b)的 Eq.（22）作为做最小化优化的错误。 因此，损失如下：

$$\mathcal{L} = -\sum_j \log\ p(y|g; W, b). \quad (24)$$

则，整个模型的损失为：

$$\mathcal{J}(\Theta) = \mathcal{L} + \frac{\lambda}{2}\|\Theta\|^2, \qquad (25)$$

其中Θ表示包含所有权重矩阵W、U和偏置向量b的模型参数，λ是正则化参数。

我们通过将误差从CRF传播到BiLSTM的隐藏层，然后通过时间反向传播(BPTT)将BiDtree CRF的所有参数从上到下更新。我们使用Adam进行梯度裁剪优化。根据经验，L2-正则化因子λ设为0.001。小批量大小为20，初始学习率为0.001。我们还在BiDtree和BiLSTM层的输出上使用dropout，dropout rate为0.5。所有权重W、U和偏置项b都是可训练的参数。 早期停止是根据验证集的性能使用的。在我们的实验中，它的价值是5个迭代。 同时，在训练过程中对初始嵌入进行微调。这意味着单词嵌入将通过反向传播梯度进行修改。 我们使用TensorFlow库，所有的计算都是在 NVIDIA Tesla K80 GPU上完成的。在算法2中总结了BiDTreeCRF的总体过程：

Algorithm 2 BiDTreeCRF Training Algorithm

Input: A set of review sentences $\mathcal{S}$ from a particular domain, $S = \{w_1, w_2, \ldots, w_i, \ldots, w_N\}$ is one of the element in $\mathcal{S}$.

Output: Learned BiDTreeCRF model

1: Construct dependency trees for each sentence S using Stanford Parser Package.
2: Initialize all learnable parameters Θ
3: **repeat**
4: Select a batch of instances $\mathcal{S}_b$ from $\mathcal{S}$
5: **for** each sentence $S \in \mathcal{S}_b$ **do**
6: Use BiDTree (1-11) to generate h
7: Use BiLSTM (14-20) to generate g
8: Compute $L(W, b)$ through (21-22)
9: **end for**
10: Use the backpropagation algorithm to update parameters Θ by minimizing the objective (25) with the batch update mode
11: **until** *stopping criteria is met*

3 实验

在本节中，我们进行了实验，以评估所提出的框架的有效性。

3.1 数据集和实验初始设置

我们使用四个基准SemEval数据集进行实验。 数据集的详细统计数据汇总在表2中。 L-14和R-14来自SemEval2014，R-15来自SemEval2015，R-16来自SemEval2016。L-14包含笔记本电脑评论，R-14、R-15和R-16都包含餐厅评论。 这些数据集正式分为三个部分：训练集、验证集和测试集。 这些分歧将被保留以进行公平的比较。 所有这些数据集都包含注释的方面术语，这些术语将用于在实验中生成序列标签。 我们使用Stanford Parser Package 生成依赖树。 评价指标为F1 score，与基本方法相同。

> 表2：SemEval的数据集；#S表示句子的数量，#T表示方面术语的数量；L-14、R-14、R-15和R-16分别表示Laptops 2014, Restaurants 2014, Restaurants 2015 and Restaurants 2016。

Datasets	Train	Val	Test	Total
L-14 $_{\#S}$	2,945	100	800	3,845
R-14 $_{\#S}$	2,941	100	800	3,841
R-15 $_{\#S}$	1,315	48	685	2,048
R-16 $_{\#S}$	2,000	48	676	2,724
L-14 $_{\#T}$	2,304	54	654	3,012
R-14 $_{\#T}$	3,595	98	1,134	4,827
R-15 $_{\#T}$	1,654	57	845	2,556
R-16 $_{\#T}$	2,507	66	859	3,432

为了初始化单词向量，我们在Amazon Re上使用基于单词包的模型(CBOW)训练单词嵌入。亚马逊评论数据集包含142.8M评论，Yelp评论数据集包含2.2M餐厅评论。 所有这些数据集都由gensim 训练，其中包含CBOW的实现。 在我们的实验中，参数*min_count*为10，*iter*为200。 我们根据在(Wang等人，2016b)中得出的结论，将单词向量的维数设置为300。 该模型的尺寸设置实验结果也表明，300是一种合适的选择，在有效性和效率之间提供了良好的权衡。

3.2 基准方法和结果

为了验证我们提出的模型在方面项提取方面的性能，我们将其与一些基准方法进行了比较：

- IHS RD，DLIREC（U），EliXa（U），和NLANGP（U）：2014年SemEval挑战赛（Chernyshevich，2014年），2014年SemEval挑战赛的R-14顶级系统（Toh和Wang，2014年），2015年SemEval挑战赛的R-15顶级系统（Vicente等人，2015年）和R-16的顶级系统分别在2016年SemEval挑战赛（Toh和Su，2016）中获奖。所有这些系统都具有相同的属性：对它们进行了各种词典和句法功能的培训，与神经网络的端到端方式相比这是劳动密集型的。U表示使用没有任何约束的其他资源，例如词典或其他培训数据。
- WDEmb：它使用词嵌入，线性上下文嵌入和依赖路径嵌入来增强CRF（Yinetal，2016）。
- RNCRF-O，RNCRF-F：他们都使用递归神经网络作为CRF输入来提取树结构特征。RNCRF-O是经过训练的无观点标签的模型。RNCRF-F不仅使用意见标签进行训练，而且还使用一些手工制作的特征进行训练（Wang等人，2016b）
- DTBCSNN + F：基于依赖树的卷积堆叠神经网络，用于捕获语法特征。其结果是由推理层产生的（Yeetal，2017）
- MIN：MIN是一个基于LSTM的深度多任务学习框架，该框架通过记忆交互共同处理方面和观点的提取任务（Li和Lam，2017年）
- CMLA，MTCA：CMLA是一个多层注意力网络，它利用方面术语和观点术语之间的关系，而无需任何解析器或语言资源进行预处理（Wang等人，2017b）。MTCA是一种多任务注意力模型，可学习不同任务之间的共享信息（Wang等人，2017a）。
- LSTM + CRF，BiLSTM + CRF：他们是（Huang et al。，2015）提出并在POS，分块和NER数据集上产生了最新的（或接近）准确性。我们将其作为ATE的基准。
- BiLSTM + CNN：BiLSTM + CNN 10是（Ma and Hovy，2016）的双向LSTM-CNNs-CRF模型。与上面的BiLSTM+ CRF相比，BiLSTM + CNN通过CNN编码了char嵌入，并在POS标记和命名实体识别（NER）的任务上获得了最新的性能。我们将此方法作为ATE的基准。CNN的窗口大小为3，过滤器的数量为30，char的尺寸为100。

> 表3：F1分数的相互比较。 '-'表明他们的论文中没有结果

Models	L-14	R-14	R-15	R-16
IHS_RD	74.55	79.62	-	-
DLIREC(U)	73.78	84.01	-	-
EliXa(U)	-	-	70.05	-
NLANGP	-	-	67.12	72.34
WDEmb	75.16	84.97	69.73	-
RNCRF-O	74.52	82.73	-	-
RNCRF+F	78.42	84.93	-	-
DTBCSNN+F	75.66	83.97	-	-
MIN	77.58	-	-	73.44
CMLA	77.80	85.29	70.73	-
MTCA	69.14	-	71.31	73.26
LSTM+CRF	73.43	81.80	66.03	70.31
BiLSTM+CRF	76.10	82.38	65.96	70.11
BiLSTM+CNN	78.97	83.87	69.64	73.36
BiDTressCRF#1	80.25	85.10	70.10	73.77
BiDTressCRF#2	80.39	**85.45**	69.83	74.01
BiDTressCRF#3	**80.66**	84.73	**70.93**	**74.53**

对于我们提出的模型，有三个变量取决于Eqs.（2-9）的权重矩阵共有与否。BiDTreeCRF#1共享所有权重矩阵，即$W_*^{\uparrow(i,o,f,u)} = W^{\uparrow(i,o,f,u)}$ 和 $U_*^{\uparrow(i,o,f,u)} = U^{\uparrow(i,o,f,u)}$，这意味着映射函数r↑(K)是无用的。BiDTreeCRF#2 共享Eqs.（2-3,5）的权重矩阵 和Eqs.（6-7,9）但不包括Eqs.（4,8）。BiDTreeCRF#3保持Eqs.（2-9）并且不共享任何权重矩阵。不同类型的权重分配意味着不同的信息传输方式。BiDTreeCRF＃1共享权重矩阵，它表示一个主词的从属词是未区分的，并且其语法关系例如 *nmod* 和 *案件*，不在考虑之列。BiDTreeCRF＃2对遗忘门的处理方式有所不同，这表明每个依赖词均受句法关系控制，以将隐藏状态传输到其下一个节点。BiDTreeCRF＃3进一步不同地对待所有门。事实证明，在句法关系控制下的精心设计的信息流是有效的。

比较结果见表3。在此表中，模型的F1得分是平均20次运行且保持相同的超参数，已在2.6节中描述，并在我们的实验中使用。我们报告用AmazonEmbedding初始化的L-14的结果。 对于其他数据集，我们使用YelpEmbedding初始化，因为它们都是餐厅评论。 我们还将在下面显示嵌入比较。

与2014年、2015年和2016年SemEvalABSA挑战的最佳系统相比，BiDTreeCRF#3在L-14、R-14、R-15和R-16上分别比IHSRD、DLIREC(U)、EliXa(U)和NLANGP(U)多获得了6.02%、0.82%、0.78%和2.15%的F1分。 具体来说，BiDTree CRF#3在L-14和R-15上的性能分别优于WDEmb5.41%和1.10%，L-14和R-14的性能分别优于RNCRF-O6.05%、2.10。 即使与利用额外手工制作特征的RNCRFF和DTBCSNNF相比，L-14上的BiDTree CRF#3和R-14上的BiDTree CRF#2没有其他语言特征(例如POS)仍然分别达到

2.15%、4.91%和0.38%、1.34%的改进。 通过内存交互训练MIN，将CMLA和MTCA设计为多任务模型，这三种方法都使用了更多的标签，并在不同的任务之间共享信息。与他们相比，BiDTreeCRF#3仍然给出了L-14和R-16的最佳分数和R-15的有竞争优势分数。而BiDTreeCRF#2实现了R-14的最高评分，尽管我们的模型被设计为一个单一的任务模型。此外，BiDTree CRF#3在所有数据集上的性能分别优于LSTMCRF和BiLSTMCRF，分别为7.14%、3.03%、4.80%和4.18%，4.47%、2.45%、4.87%和4.38%，这些改进是显著的($p<0.05$)。 考虑到BiLSTMCRF可以看作是没有BiDTree层的BiDTreeCRF#3，所有的结果都支持BiDTree可以有效地提取句法信息。

正如我们所看到的，所提出的模型的不同变体在四个数据集上具有不同的性能。 特别是，BiDTreeCRF#3比L-14上的其他变体更强大，R-15、R-16和BiDTreeCRF#2 对R-15更有效。 我们认为，R-15是一个带有一些"NULL"方面术语的小数据集，这是这些baseline的性能在它们之间有很小差距的原因。 证明了它是一个很难提高分数的数据集。 因此，这是一个鼓舞人心的结果，尽管BiDTreeCRF#3在没有其他辅助信息（例如意见术语）的情况下比MTCA稍差。 此外，BiDtreeCRF#3即使没有字符嵌入，也优于BiLSTMCNN。 请注意，我们没有为实际目的对BiDTreeCRF的超参数进行调优，因为这个调优过程是耗时的

3.3 Ablation实验

> 表4：BiDTreeCRF 的消融实验的F1评分.

Models	L-14	R-14	R-15	R-16
BiLSTM+CRF	76.10	82.38	65.96	70.11
BiDTree+CRF	71.56	81.36	64.09	67.56
DTree-up	78.66	84.32	69.35	72.10
DTree-down	78.36	84.63	68.69	72.34
BiDTreeCRF#3	**80.69**	**84.89**	**71.24**	**74.56**

为了测试BiDTreeCRF各组分的影响，对BiDTreeCRF#3不同层进行了以下烧蚀实验：（1）DTree-up：BiDTree自下而上的传播连接到BiLSTM和CRF层。（2）DTree-down：Bi DTree的自顶向下传播连接到Bi LSTM和CRF层。（3）Bi DTree+CRF：Bi LSTM层与Bi DTree CRF相比没有使用。 初始词嵌入与以前相同对比结果见表4。 将BiDTree CRF与DTree-up和DTree-down进行比较，可以明显看出BiDTree比任何单一方向依赖网络都更具有竞争力，这是提出的BiDTree CRF的原始动机。 BiDTree CRF优于BiDTree CRF的事实表明，BiLSTM层在BiDTree的顶部提取序列信息是有效的。 另一方面，BiDTree CRF优于BiLSTM CRF的事实表明，BiDTree提取的依赖句法信息在方面项提取任务中是非常有用的。 以上所有改进均有显着性差异($p<0.05$)，并采用统计t检验。

3.4 单词嵌入与句法关系

由于单词嵌入是学习数据较少的重要因素，我们还对单词嵌入进行了比较实验。 此外，句法关系Eqs.（2-5）也作为比较标准。 实验设置:如小批量大小和学习率，与以前的设置相同，除了单词嵌入和不集成句法关系知识之外，没有其他变化。

> 图6：Amazon Embedding vs.Yelp Embedding(E-Amazon vs.E-Yelp)，带句法关系。

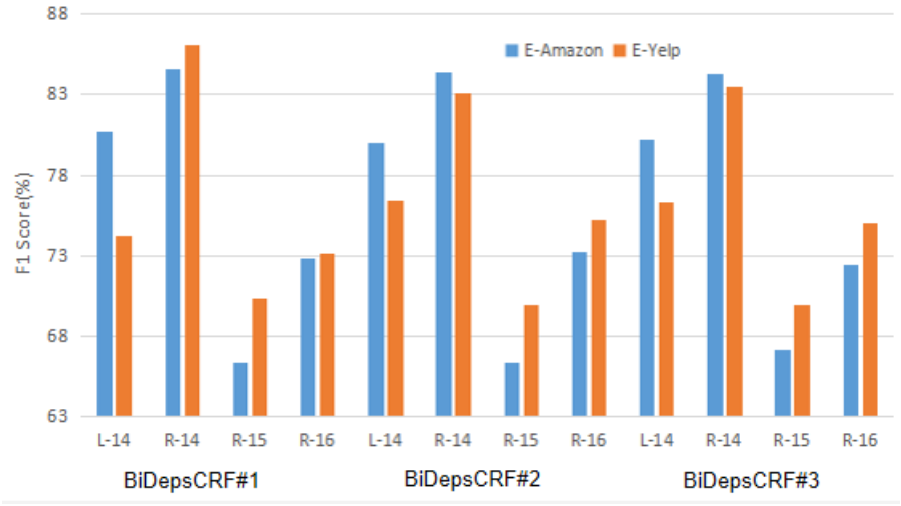


图7：Amazon Embedding vs.Yelp Embedding(E-Amazon vs.E-Yelp)，不含句法关系。

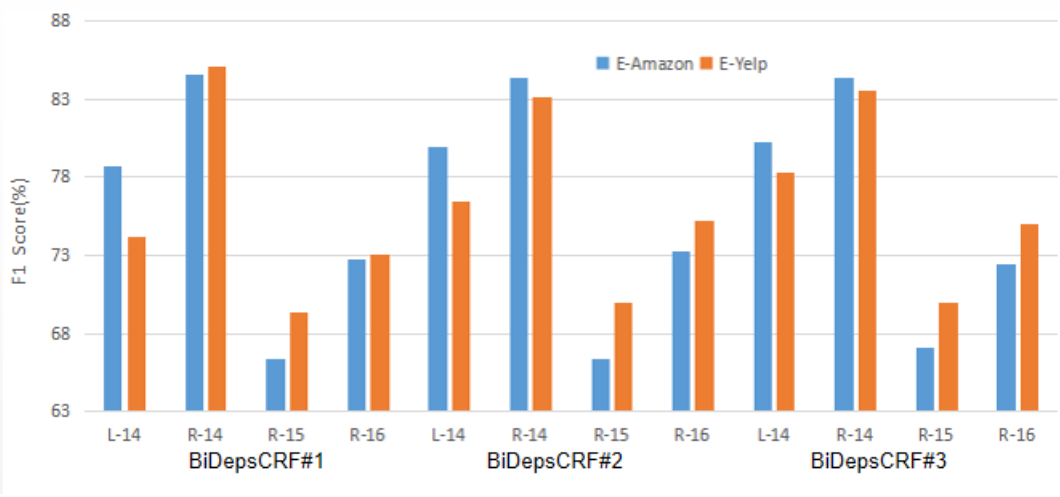


图6和图7说明了AmazonEmbedding和YelpEmbedding之间的比较。 每个数字涉及四个数据集上的BiDTreeCRF的三个变体。 所有这些都表明，AmazonEmbedding总是优于L-14的YelpEmbedding，而YelpEmbedding比R-14、R-15和R-16的AmazonEmbedding具有绝对优势。 YelpEmbedding是*restaurant*的域内嵌入，AmazonEmbedding是*laptop*的域内嵌入，这表明域内嵌入比域外嵌入更有效。

图8：带句法关系vs不带句法关系(with-REL vs No-REL) 使用Amazon Embedding

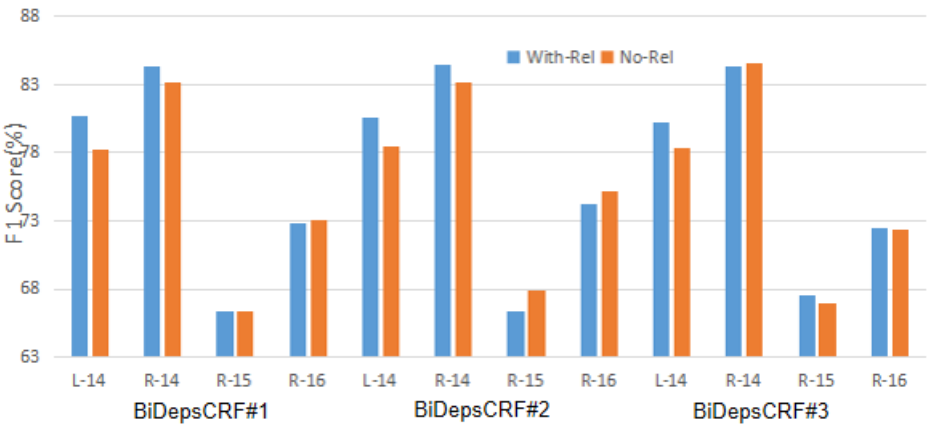


图9：带句法关系vs不带句法关系(with-REL vs No-REL) 使用 Yelp Embedding

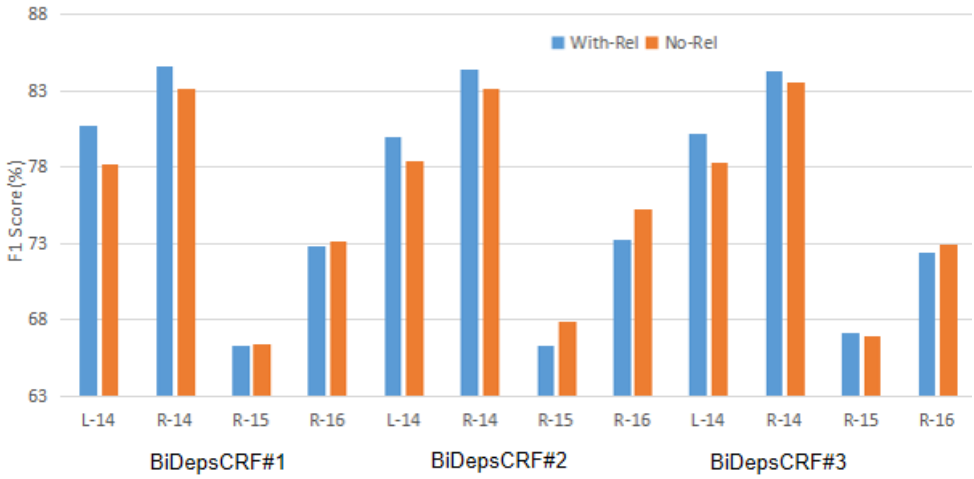
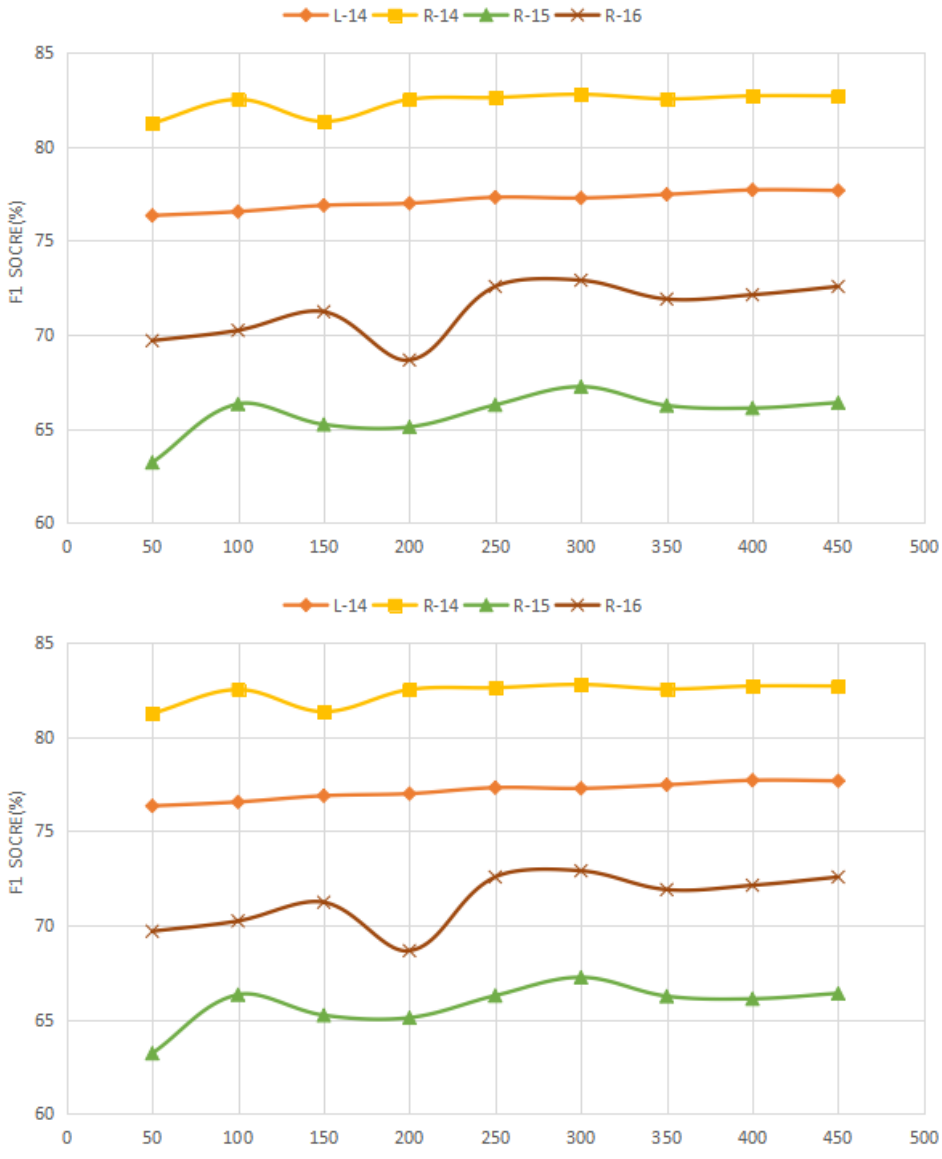


图8和图9显示了不同句法关系条件的比较。 图8是使用AmazonEmbedding进行的比较，图9是使用YelpEmbedding进行的比较。与无句法关系的模型相比，句法关系信息有助于提高性能。

3.5 灵敏度测试

> 图10：单词嵌入的敏感性研究。 顶部：F1分的BiDTreeCRF#3与不同的词向量维度d 在Electronics Amazon Embedding。 底部：BIDTreeCRF#3 在Yelp Embedding上具有不同词向量维数d的F1评分。



对BiDTree CRF#3词嵌入维数d进行了灵敏度测试。涉及到不同的尺寸(从50到450不等，增量为50。图10分别使用AmazonEmbedding和YelpEmbedding给出了四个数据集上的灵敏度图。 值得一提的是，亚马逊嵌入这里只是从电子产品的评论，考虑到时间成本。 虽然分数略低于从整个亚马逊评论语料库训练的嵌入，但结论仍然成立。图表明，300是所提出的模型的合适尺寸。 也证明了我们的模型的稳定性和鲁棒性。

3.6 案例研究

> 表5：BiDTreeCRF和BiLSTM的提取比较

Text (The ground-truth of aspect terms is marked with bold font)	**Dependency Relationships**	**BiDTreeCRF**	**BiLSTM**
Other than not being a fan of **click pads** (industry standard these days) and the lousy **internal speakers**, it's hard for me to find things about this notebook I don't like, especially considering the $350 **price tag**.	$click \xleftarrow{compound} pads$, $internal \xleftarrow{amod} speakers$, $price \xleftarrow{compound} tag$	click pads, internal speakers, price tag	internal speakers, price tag
Keyboard responds well to presses.	$Keyboard \xleftarrow{nsubj} responds$	Keyboard	Keyboard, responds
I am please with the products ease of **use**; out of the box ready; **appearance** and **functionality**.	$ease \xrightarrow{nmod} use \xrightarrow{case} of$, $appearance \xrightarrow{cc} and$, $appearance \xrightarrow{conj} functionality$	use, appearance, functionality	use, functionality
With the **softwares** supporting the use of other **OS** makes it much better.	$use \xrightarrow{nmod} OS \xrightarrow{case} of$, $the \xleftarrow{det} softwares$, $softwares \xleftarrow{nsubj} supporting$	softwares, OS	softwares, use, OS
I tried several **monitors** and several **HDMI cables** and this was the case each time.	$monitors \xrightarrow{cc} and$, $monitors \xrightarrow{conj} cables$, $cables \xrightarrow{compound} HDMI$	monitors, HDMI cables	HDMI cables

表5显示了L-14数据集的一些例子，以证明BiDTreeCRF的有效性。 第一列包含评论，相应的方面术语用粗体字体标记。 第二列描述了与方面术语相关的一些依赖关系。 第三列和最后一列分别是BiDTreeCRF和BiLSTM的提取结果。总体来说BiDtreeCRF能比BiLSTM更好地提取方面术语，遗漏和错误较少。 在第一个例子中，BiLSTM忽略了方面术语*"click pads"*，但其内部关系类似于$price \xleftarrow{compound} tag$，这在BiD Tree CRF中可以被认为是一个重要的特征。 因此，双DTreeCRF可以准确地提取它。同样，通过关系$Keyboard \xleftarrow{nsubj} responds$，BiDTreeCRF可以避免将 *"responds"* 作为一个方面术语。 对于第三个例子和第四个例子中的同一个词 *"use"* ，一个是真正的方面术语，另一个不是。原因反映在这两个关系中：$ease \xrightarrow{nmod} use \xrightarrow{case} of$ 和$use \xrightarrow{nmod} OS \xrightarrow{case} of$。 对于最后的例子 *"monitors"* 和 *"cables"* 是等价关系，因为$monitors \xrightarrow{conj} cables$，因此，它们是由BiDtree CRF同时提取的，而不是由BiLSTM只提取其中的一部分。 所有上述分析都提供了支持证据，证明我们在依赖树上构建的BiDTree CRF是有用的，并且可以利用单词之间的关系来提高ATE的性能。

4 相关工作

情感分析作为一个重要而实用的课题，在文献中得到了广泛的研究(Hu and Liu, 2004; Cambria, 2016)，特别是ATE。 解决ATE问题有几种主要的方法。Hu和Liu(2004)使用频繁模式挖掘方法提取了经常出现的名词和名词短语的方面词。 邱等人(2011)和Liu等人(2015b)提出了一种基于规则的方法，基于观点或情感必须有目标的思想，利用关于方面术语(也称为目标)和情感词之间的一些句法关系的手工或自动生成规则(Liu, 2012)。Chen等人(2014)采用主题建模来解决ATE问题，该模型采用了一些基于潜在Dirichlet分配(LDA)的概率图形模型(Bleietal ，2003)及其变体。 以上方法都是基于无监督学习的。 对于监督学习，ATE主要被认为是一个顺序标注问题，并通过隐马尔可夫模型(Jinetal, 2009)或CRF来解决。然而，传统的监督方法需要设计一些词汇和人为地使用语法特性来提高性能。 神经网络是解决这一问题的有效途径。

最近的研究表明，神经网络确实可以在ATE中取得有竞争力的好成绩。 Irsoy和Cardie(2013)应用深度Elman-type递归神经网络(deep Elman-type Recurrent Neural Network, RNN)提取意见表达，结果表明深度RNN优于CRF、半CRF和浅层RNN。 Liu等人(2015a)进一步实验了更先进的RNN。此外，他们指出，使用其他语言特征(如POS)可以得到更好的结果。 与这些文章不同的是，Poria等人(2016)使用了7层深度卷积神经网络(CNN)在有主见的句子中为每个单词标注一个方面或非方面标签。一些语言模式也被用来提高标记的准确性。 注意机制和记忆交互作用也是ATE的有效方法。 Li and Lam(2017)采用两个LSTMs通过记忆交互共同处理提取任务。 这些LSTMs配备了扩展记忆和神经记忆操作。 Wangetal. (2017b)提出了一种多层次注意网络来处理方面术语的共提取任务，利用术语之间的间接关系来更精确地提取信息。 He等人(2017)提出了一种无监督神经注意力模型来发现相干方面。 它的核心思想是通过神经词嵌入来挖掘词的共现分布，并在训练过程中使用注意机制去强调不相关的词。 然而，基于句子序列结构的RNN和CNN不能有效地、直接地捕捉到能更好地反映自然语言句法特性的基于树的句法信息，因此对ATE算法非常重要。

研究人员已经提出了一些基于树的神经网络。 例如，Yin等人(2016)设计了一种既考虑线性上下文又考虑依赖上下文信息的词嵌入方法。 由此产生的嵌入被用于CRF中提取方面术语。 该模型证明，词汇间的句法信息比其他具有代表性的词汇间的句法信息具有更好的性能。 但是，它涉及两个阶段的过程，而不是一个端到端系统直接从依赖路径信息训练到最终的ATE标签。 相反，我们提出的BiDTreeCRF是一个端到端的深度学习模型，它不需要任何手工制作的特性。 Wang等人(2016b)将依赖树和CRF集成到一个统一的框架中，用于明确的方面和观点术语共提取。 然而，依赖树上的单向传播不足以表示完整的树结构语法信息。 我们采用双向传播机制来提取信息，而不是依赖树各层的全连接，这在我们的实验中被证明是有效的。 Ye et al.(2017)提出了一种基于树的卷积来捕捉句子的句法特征，这使得保持顺序信息变得困难。 我们融合了树形结构和顺序信息，而不是仅仅使用单一的表示来有效地解决ATE。

本文还涉及到其他几个基于群体树构建的模型，这些模型用于完成其他一些NLP任务，如翻译(Chenetal, 2017)、关系提取(Miwa and Bansal, 2016)、关系分类(Liuetal，2015c)和语法语言建模(Taietal，2015；滕和张，2016；张等，2016)。 然而，我们有不同的模型和不同的应用。

5 结论

本文介绍了端到端的BiDTreeCRF框架。 该框架通过自底向上和自顶向下的依赖树传播，有效地提取依赖语法信息。 通过将依赖语法信息与BiLSTM和CRF的优点相结合，我们在不使用任何其他语言特征的情况下，在四个基准数据集上实现了最先进的性能。 已对提议模型的三种变体进行了评估，并表明它们比现有的最先进的基线方法更有效。 这些变体的区别取决于它们在训练过程中是否具有相同的重量。 我们的结果表明，依赖句法信息也可以用于方面术语和方面意见的共提取，以及其他序列标注任务。 额外的语言特征(如POS)和字符嵌入可以进一步提高所提模型的性能。

注：实验结果处理成图片替代了原始论文的图表