

句法增强的 UCCA 语义分析方法

蒋炜 李正华[†] 张民

苏州大学计算机科学与技术学院, 苏州 215006; [†] 通信作者, E-mail: zhli13@suda.edu.cn

摘要 普适概念认知标注 (Universal Conceptual Cognitive Annotation, 简称 UCCA) 是一种简单、多语言通用的语义表示形式。考虑到句法结构与语义结构之间的紧密联系, 尝试将句法信息融入到 UCCA 语义分析模型中。基于目前性能最好的基于图的 UCCA 语义分析模型, 提出并比较了四种不同的融入依存句法信息的方法。采用 SemEval-2019 国际评测语义分析任务的英文数据集进行了实验, 在本领域和跨领域两个数据集上的结果均表明, 句法增强的方法能够显著提高语义分析性能。引入 BERT 特征后, 句法信息仍然可以提供一定的帮助。
关键词 语义分析; UCCA; 句法分析

Syntax-enhanced UCCA Semantic Parsing

JIANG Wei, LI Zhenghua[†], ZHANG Min

School of Computer Science and Technology, Soochow University, Suzhou 215006;

[†] Corresponding author, E-mail: zhli13@suda.edu.cn

Abstract Universal Conceptual Cognitive Annotation (UCCA) is a simple and cross-linguistically applicable semantic representation scheme. Considering the close correlation between syntactic and semantic structures, this paper attempts to add syntactic information into the UCCA semantic parsing model. Based on the state-of-the-art graph-based UCCA semantic parser, the authors propose and compare four different approaches for incorporating syntactic information. The authors conduct experiments on the English benchmark dataset for the semantic parsing shared task of the SemEval-2019 conference. The results on both the in-domain and out-domain evaluation data show that syntax-enhanced methods can significantly improve performance of semantic parsing. After utilizing BERT, syntactic information is still beneficial to some extent.

Key words semantic parsing; UCCA; syntactic parsing

普适概念认知标注 (Universal Conceptual Cognitive Annotation, 简称 UCCA) 是一种近年提出的多语言通用的语义表示形式 (Abend and Rappoport, 2013^[1])。和其他语义表示形式如 AMR (Abstract Meaning Representation) 相比, UCCA 具有形式简洁 (标签少)、易于理解的特点, 从而不需要语言学专家参与就可以实现快速标注 (Abend et al., 2017^[2])。Birch et al. (2017)^[3] 发现 UCCA 语义结构可以有效帮助机器翻译结果评价任务。

如图 1 所示, UCCA 采用有向无环图 (Directed Acyclic Graph, 简称 DAG) 来表示句子的语义结构。图中的每个叶子节点依次对应句子中的所有单词, 每个非叶子节点对应一个单独的语义实体。非叶子节点之间的边上都带有类别标签, 代表了孩子节点在父亲关系中扮演的角色。图 1 给出了一个英文的 UCCA 示例。它的中文意思是“毕业后, John 放弃了一切”, 其中包含了两个主要场景, 分别为“John 毕业”和“John 放弃了一切”, 对应图中的节点 3 和节点 2。UCCA 图中的一个节点可能有多个父亲, 例如图 1 中的节点 8, 其中一个父亲节点用主边 (Primary Edge) 连接, 即实线边 $8 \leftarrow 2$, 而其他父亲节点都用远程边 (Remote Edge) 连接, 即虚线边 $8 \leftarrow 3$ 。所有的主边形成一个树状的结构, 而远程边导致的重入性形成了 DAG。UCCA 的另一个特征是存在不连续节点, 例如图 1 中的节点 4, 叶子节点 everything 在它的覆盖范围但不是它的子孙。

考虑到 UCCA 的这种结构, Hershovich et al. (2017)^[4] 首先提出了基于转移的 UCCA 分析模型 TUPA。

在前人的工作基础上，TUPA 增加了 swap 动作来解决不连续结构并且区分主边与远程边。Jiang et al. (2019)^[5] 则提出了一种基于图的 UCCA 分析模型，他们将 UCCA 图通过规则转化为短语结构树，并且在标签上添加额外的信息用于将来的还原，实验结果表明该模型比 TUPA 更加有效。之后 Hershcovich et al. (2018)^[6] 又扩展了他们自己的工作，将 UCCA 与 AMR、SDP (Semantic Dependency Parsing) 等其他语义分析进行多任务学习 (Multi-task Learning, 简称 MTL)。其中 UCCA 作为主任务而其它语义分析作为辅助任务，由此构建了一个通用的基于转移的多任务学习模型，表明了其他语义任务对于 UCCA 有着不可忽视的辅助作用。

除了不同的语义表示之间有着共性，过去的工作还表明句法和语义是互补互助的。Xia et al. (2018)^[7] 在语义角色标注任务中尝试了多种不同方式产生句法特征，直接作为编码器的额外输入，结果表明了句法对浅层语义分析的有益性。江腾蛟等^[8] 提出了语义角色标注与依存句法分析相结合的评价对象-情感词对抽取规则，张志远等^[9] 尝试了基于语义和依存句法特征的评论对象抽取方法，实验结果都表明了两者结合的有效性。

事实上 UCCA 与句法结构之间也有很强的联系，比如依存句法树的树根往往对应了 UCCA 图中的一个场景谓词。所以本文在 Jiang et al. (2019)^[5] 的工作基础上，首次提出并尝试了四种不同的方法利用依存句法分析来提高 UCCA 语义分析的性能。英文数据上的实验结果表明句法信息能有效地帮助 UCCA 语义分析，而融入了 BERT^[10] 之后，句法模型产生的显示句法信息就明显弱化，只能提供有限的帮助。

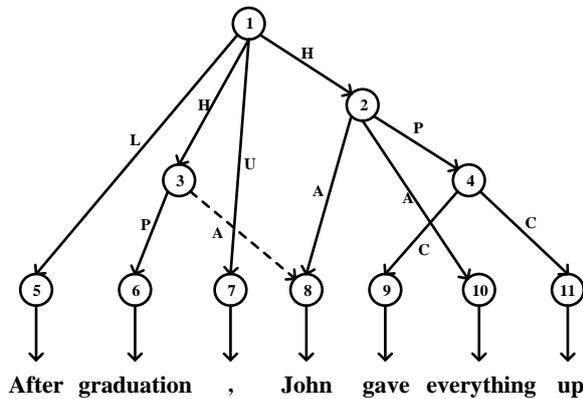


图 1 UCCA 示例。为了方便说明，我们给每个非叶子节点都编了号。

Fig.1 An Example of UCCA. To facilitate illustration, we assign a number to each non-terminal node.

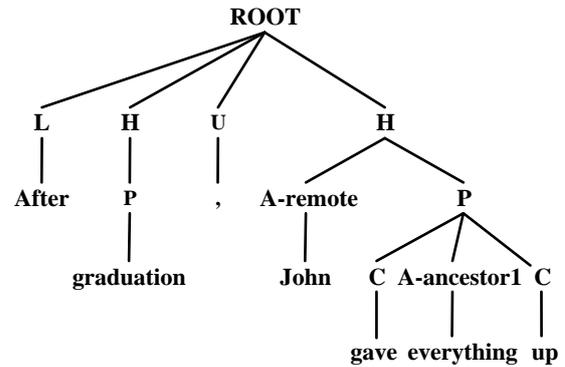


图 2 由 UCCA 图转化而来的短语结构树。

Fig.2 Constituent tree converted from UCCA graph.

1 UCCA 语义分析基准模型

Jiang et al. (2019)^[5] 提出了一种简洁有效的方法用于分析 UCCA 图，核心思想是消除图中的远程边和不连续节点，并且添加额外的标签用于之后的恢复，将其转化为标准的短语结构树。给定图 1 中的 UCCA 图，他们通过以下算法将其转化为图 2 中的短语结构树：

- (1) 移除远程边。对于那些拥有多个父亲的节点，移除指向它的远程边，只保留主边。为了之后的恢复，在主边的标签上添加额外的“-remote”标记，表明该节点还有其它的父亲节点与它有远程关系。
- (2) 消除不连续节点。图 1 中的 4 号节点被称为不连续节点，因为它的叶子子孙并非连续。由于主流的短语结构树分析器不能解决类似情况，所以需要移动特定的边来消除这种不连续的结构。给定一个不连续节点 $A = 4$ ，首先处理最左侧的非子孙叶子 $B = \text{everything}$ ，从 B 出发依次向上遍历祖先节点，直到找到一个节点 $C = 10$ ，它是一个不连续节点，或者它的父亲是 A 和 B 的最小公共祖先。记 $C = 10$ 的父亲为 $D = 2$ 。然后移动边 $D \rightarrow C$ 使 C 成为 A 的孩子，同时在该边的标签上添加额外的“ancestor 1/2/3...” 或者 “discontinuous”，其中数字表示 $D = 2$ 到 $A = 4$ 经过的边数。经过此次移动后，叶子节点 everything 成为了 A 的子孙。重复上述操作直到图中的不连续结构都消除。

(3) 将边上的标签移动到节点上。由于结构树中的标签都对应每个节点而不是边，必须将 UCCA 图中边的标签都转移到它的孩子节点上，同时给予根节点“ROOT”标签。

Jiang et al. (2019)^[5]直接采用了 minimal span-based parser(Stern et al., 2017^[11])作为短语结构树分析模型。给定输入句子 $S = w_1, w_2, \dots, w_n$ ，每个词 w_i 都转化为一个高维度的稠密向量：

$$x_i = e_{w_i}^{word} \oplus e_{w_i}^{pos} \oplus \dots, \quad (1)$$

其中 $e_{w_i}^{word}$ 是通过查表操作获得的词向量， $e_{w_i}^{pos}$ 则是 w_i 词性对应的向量。除此之外，Jiang et al. (2019)^[5]还利用了语料中提供的自动依存标签^①和自动实体标签^②，但是发现带来的提升非常有限，所以本文中舍弃了这些自动特征。然后多层 BiLSTM 对这些词表示向量进行编码，将最后一层的输出 $[f_i^U; b_i^U]$ 取出，则 span 的表示为：

$$r_{i,j} = (f_j^U - f_i^U) \oplus (b_i^U - b_j^U), \quad (2)$$

其中 f_i^U 和 b_i^U 分别是第 i 个词对应最后一层正向和反向 LSTM 的输出向量。最后这些 span 表示输入到 MLP 中来计算 span 分割和标签的分数：

$$S_{split}(i, j), S_{labels}(i, j) = MLP^s(r_{i,j}), MLP^l(r_{i,j}). \quad (3)$$

预测时，分析模型根据这两个分数贪心地自顶向下搜索来构建树(Stern et al., 2017^[11])。

为了从预测出的树还原到 UCCA 图，上一小节中的三个步骤需要被逆向地执行，即把标签移动到边上、根据额外标签“-ancestor”向上移动边的一端以及恢复远程边。前两步都可以根据规则来完成，但恢复远程边需要额外的模型来处理。给定一个远程节点 A （即标签带有“-remote”的边所指向的节点）和候选父亲节点 B （除了 A 及其父亲的其他所有非叶子节点），我们直接预测 B 到 A 的关系类别，如果它们之间的关系标签是人为添加的“NOT-PARENT”，则判断 B 不是 A 的父亲也就是没有远程边，否则认为存在一条远程边 $A \leftarrow B$ 。至于如何去预测 B 到 A 的关系标签，我们借鉴了目前效果最好的句法分析模型 Biaffine Parser(Dozat and Manning^[12])，假定它们叶子子孙的起始位置分别为 i, i' 和 j, j' ，首先用 span 向量 $r_{i,j}$ 和 $r_{i',j'}$ 分别表示它们，然后经过两个 MLP 产生 $r_{i,j}^{child}$ 和 $r_{i',j'}^{parent}$ ，最后用双仿射操作得到一个标签分数向量，分值最大标签即为预测结果：

$$score(A \leftarrow B) = \left[r_{i,j}^{child} \right]_1^T W_r r_{i',j'}^{parent}, \quad (4)$$

它的维度是标签集合的大小，包括人为添加的空标签。最终结果中，如果预测出的不是“NOT-PARENT”标签，则添加对应的远程边。该模型与短语结构分析器置于多任务学习框架下一起训练，共享相同的输入以及编码器，如图 4 中左虚线框内所示，其中 MLP&Biaffine 就是用来恢复远程边（Remote Recovery）的子模型，而右侧的 MLPs 则是用于解析短语结构树（Constituent Parsing）。

训练时，将所有远程节点与候选父亲节点之间的交叉熵损失累加起来，再加上分析树的最大边缘损失^[11]作为整个多任务学习框架的损失函数。

2 依存句法分析模型

为了产生句法信息，我们选取了目前性能最好的 Biaffine Parser (Dozat and Manning^[12])作为基础句法模型，用它生成自动句法分析结果或者词表示向量。Biaffine Parser 利用深度双仿射操作计算所有依存弧的分值，然后用最小生成树解码算法得到一棵分值最大的树。如图 4 右侧虚线框内所示。首先，对于一个待分析的句子 $S = w_1, w_2, \dots, w_n$ ，Biaffine Parser 采用 L 层 BiLSTM 来编码句子的信息，最底层 BiLSTM 第 k 个位置的输入为词 w_k 的词向量 $e_{w_k}^{word}$ 及词性向量 $e_{w_k}^{pos}$ 的拼接向量 x_k ，与 UCCA 分析模型类似。最顶层 BiLSTM 第 k 个位置的输出为 h_k （即前向 LSTM 输出和后向 LSTM 输出的拼接）。这些隐层表示输入到两个单独的 MLP 映射到低维度的词表示，分别作为核心词和修饰词：

$$h_k = f_{k,L}^B \oplus b_{k,L}^B, \quad (5)$$

^① 词性和依存标签由 <http://ufal.mff.cuni.cz/udpipe> 提供

^② 命名实体标签标签由 <http://spacy.io> 提供

$$r_i^D, r_j^H = \text{MLP}^D(h_i), \text{MLP}^H(h_j). \quad (6)$$

最后采用双仿射运算，计算出弧 i 到 j 的得分：

$$\text{score}(i \leftarrow j) = \begin{bmatrix} r_i^D \\ 1 \end{bmatrix}^T W^b r_j^H. \quad (7)$$

类似地，依存弧标签的分值 $\text{score}(i \leftarrow j)$ 同样用额外的 MLP 和双仿射操作来计算。对于每个词 w_i 及其正确的依存弧 w_j 和标签 l ，损失函数为局部的交叉熵函数之和。

3 句法增强的 UCCA 语义分析

直觉上讲，句法和语义有着密不可分的关系。从结构上看，UCCA 也与依存句法息息相关，比如依存树中名词主语标签所指向的节点对应了 UCCA 图中的一个参与者 (Participate)，就像图 1 中的节点 8 和图 3 中的 John 节点；依存树中动词短语也会对应 UCCA 图中的一个节点，就像图 3 中 gave up 之间的标签 prt (phrasal verb particle) 就代表这两个词是一个短语，正好对应了图 1 中的 4 号节点。因此我们尝试用依存句法分析来增强语义分析的效果。上两节中我们分别介绍了基础的 UCCA 分析模型与句法分析模型，我们记 x_i^{syn} 为句法特征向量，它被拼接在 UCCA 模型的输入 x_i 之后即：

$$x_i = x_i \oplus x_i^{\text{syn}}. \quad (8)$$

下面分别介绍四种不同的方法利用依存句法生成 x_i^{syn} 。

3.1 基于依存标签嵌入的方法

依存句法树蕴含着丰富的句法信息，最简单的用法就是加入依存标签。我们尝试利用预训练好的 Biaffine Parser 预测出输入句子的依存句法树，和词性标签一样，将每个词对应的依存标签映射到一个固定维度的向量，拼接在词表示向量之后，并且随着训练一起微调。如公式 9 所示：

$$x_i^{\text{syn}} = e_{w_i}^{\text{label}}. \quad (9)$$

3.2 基于 Tree-GRU 句法树编码的方法

相比只用依存标签，树状递归神经网络 (Tree-structured Recurrent Neural Network, 简称 Tree-RNN) (Tai et al., 2015^[13]; Miwa et al., 2016^[14]) 能够全局地编码整棵依存树，并且得到每个词在整棵树中对应的位置信息和句法特征。过去的许多工作表明，树状的递归神经网络能更有效地帮助关系抽取 (Miwa and Bansal, 2016^[14]; Feng et al., 2017^[15])、机器翻译 (Chen et al., 2017^[16]) 等任务。我们使用类似 Chen et al. (2017)^[16] 的双向 Tree-GRU (Tree-structured Gated Recurrent Unit) 来编码预测出来的依存树。其中自底向上的 Tree-GRU 根据每个节点的输入和它的所有孩子来计算隐状态，如图 3 中的红色线，公式 10 表明了详细计算过程：

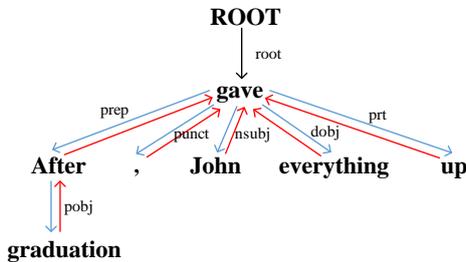


图 3 双向的 Tree GRU

Fig.3 Bi-directional Tree GRU

$$\begin{aligned} \bar{h}_{i,L}^\uparrow &= \sum_{j \in \text{lchild}(i)} h_j^\uparrow, \bar{h}_{i,R}^\uparrow = \sum_{k \in \text{rchild}(i)} h_k^\uparrow \\ r_{i,L} &= \sigma(W^{rL} e_{w_i}^{\text{label}} + U^{rL} \bar{h}_{i,L}^\uparrow + V^{rL} \bar{h}_{i,R}^\uparrow) \\ r_{i,R} &= \sigma(W^{rR} e_{w_i}^{\text{label}} + U^{rR} \bar{h}_{i,L}^\uparrow + V^{rR} \bar{h}_{i,R}^\uparrow) \\ z_{i,L} &= \sigma(W^{zL} e_{w_i}^{\text{label}} + U^{zL} \bar{h}_{i,L}^\uparrow + V^{zL} \bar{h}_{i,R}^\uparrow) \\ r_{i,R} &= \sigma(W^{zR} e_{w_i}^{\text{label}} + U^{zR} \bar{h}_{i,L}^\uparrow + V^{zR} \bar{h}_{i,R}^\uparrow) \\ z_i &= \sigma(W^z e_{w_i}^{\text{label}} + U^u \bar{h}_{i,L}^\uparrow + V^z \bar{h}_{i,R}^\uparrow) \\ \hat{h}_i^\uparrow &= \tanh(W e_{w_i}^{\text{label}} + U(r_{i,L} \odot \bar{h}_{i,L}^\uparrow) + V(r_{i,R} \\ &\quad \odot \bar{h}_{i,R}^\uparrow)) \\ h_i^\uparrow &= z_{i,L} \odot \bar{h}_{i,L}^\uparrow + z_{i,R} \odot \bar{h}_{i,R}^\uparrow + z_i \odot \hat{h}_i^\uparrow \end{aligned} \quad (10)$$

其中 $\text{lchild}/\text{rchild}(\cdot)$ 表示左/右孩子集合。类似地，自顶向下的 Tree-GRU 则根据各个节点的父亲来计算隐状态，由于每个节点只可能有一个父亲，所以该过程类似于常规的 LSTM，如图 3 中的蓝色线。最后将两

个方向上的隐层状态拼接起来得到每个词的句法特征表示：

$$\mathbf{x}_i^{syn} = \mathbf{h}_i^{\uparrow} \oplus \mathbf{h}_i^{\downarrow}. \quad (11)$$

3.3 基于句法感知预训练词表示的方法

Peter et al. (2018)^[17]提出了一种通用的预训练语言模型 ELMo 用于产生底层的特征表示作为下游模型的输入。这些基于上下文的向量蕴含了丰富的文本信息能有效地帮助许多 NLP 任务。同样的，除了依存树，句法分析模型的编码器输出也蕴含了许多句法信息，最近的研究表明它对机器翻译也有巨大帮助 (Zhang et al., 2019^[18])。因此我们结合两者的思想用预训练好的 Biaffine Parser 来产生类似的特征向量作为 UCCA Parser 的额外输入，各层的 BiLSTM 输出经过加权求和并乘上一个特定系数后直接拼接在 \mathbf{x}_i 之后，如公式 12：

$$\mathbf{x}_i^{syn} = \gamma \sum_{j=0}^L s_j (\mathbf{f}_{ij}^B \oplus \mathbf{b}_{ij}^B), \quad (12)$$

其中 L 代表了 Biaffine Parser LSTM 的层数， γ 和 s_j 都是模型的参数，随着训练调整，而 Biaffine Parser 的参数不再微调。通过这种方式能充分利用编码器不同层次的信息，同时让模型自动地平衡不同层次信息对于语义分析任务的贡献。

3.4 基于句法感知微调词表示的方法

上一节的方法中，句法产生的词表示在训练语义分析模型时不会微调，这导致语义分析模型与句法模型的耦合性比较低，两者可能存在更佳的合作方式。因此，我们尝试同时训练 UCCA Parser 与 Biaffine Parser，从而微调句法感知的词表示。该模型如图 4 所示，左侧为语义分析模型而右侧为句法分析模型。

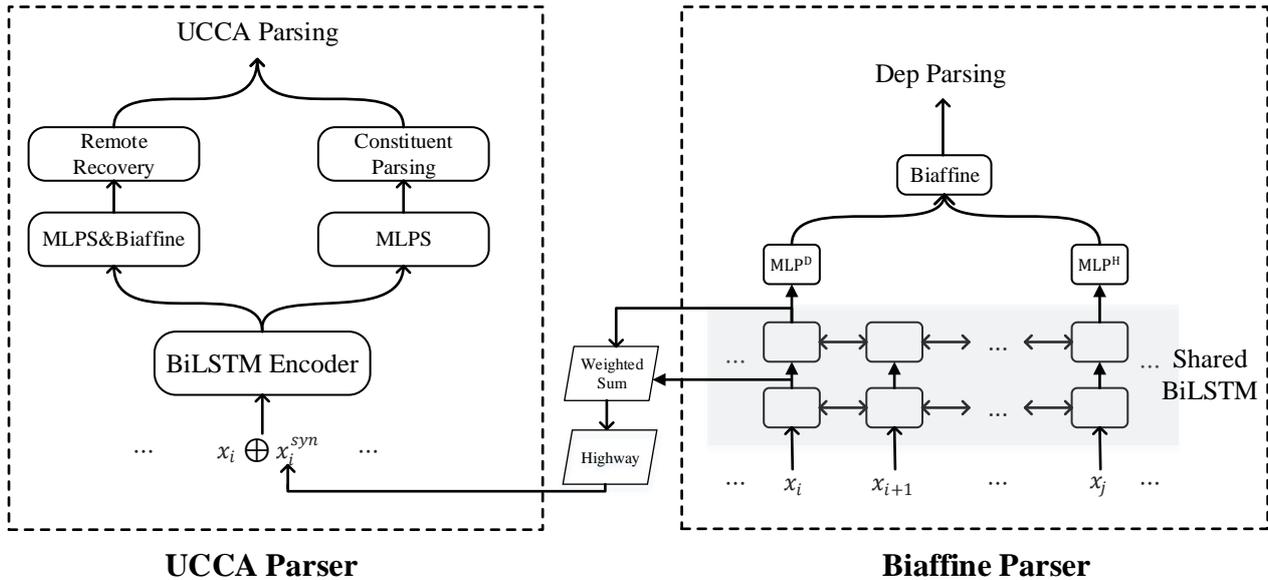


图 4 基于句法感知微调词表示方法的框架图。左虚线框内为 UCCA 语义分析模型，右虚线框内为句法分析模型。

Fig.4 The framework of syntax-aware fine-tuning word representation approach. UCCA semantic parser is in the left dotted box and dependency parser is in the right dotted box.

当要分析句法时，输入的句子仅仅通过 Biaffine Parser 部分，所以句法模型的训练不会影响到左侧的语义分析模型。而要分析 UCCA 时，输入的词表示向量首先通过 Biaffine Parser 的 BiLSTM，然后将其各层的输出取出，做一次加权求和，并乘上一个特定任务系数，最后经过一层 Highway，目的是将词表示映射到不同的语义空间：

$$\mathbf{o}_i^{syn} = \gamma \sum_{j=0}^L s_j (\mathbf{f}_{ij}^B \oplus \mathbf{b}_{ij}^B), \quad (13)$$

$$\mathbf{x}_i^{syn} = \text{Highway}(\mathbf{o}_i^{syn}). \quad (14)$$

得到的句法特征向量 \mathbf{x}_i^{syn} 拼接在每个词表示 \mathbf{x}_i 上，再把该扩展的词表示输入到 UCCA Parser 的 BiLSTM 中，

最终分析出 UCCA 图。这样两个模型共享的不仅是最底层的词表示向量，还有句法模型的编码器，当 UCCA 的损失函数进行梯度回传时，句法模型的编码器也会受到影响，使其同时理解句子的语义信息，从而微调产生的 x_i^{syn} 。我们没有设计成常规的 MTL 框架，比如两个模型直接共享一个编码器或者像 Hershcovich et al. (2018)^[6]一样把多个任务的编码器并行组合是因为：1) 从任务本身来讲，句法是理解语义的基础，句法可能更适合作为理解语义的先验任务。2) 从抽象出来的任务来讲，句法旨在分析词与词之间的关系，而基于图的 UCCA Parser 旨在分析 span 与 span 的关系，比词级别更高层次，因此我们将句法模型置于语义分析模型的先验位置。

4 实验

实验数据。我们在 1.2.4 版本的 UCCA 英语维基百科语料(English Wikipedia corpus, 简称 English-Wiki) 上进行实验，并且使用 1.2.2 版本的 UCCA 《海底两万里》英法平行语料 (Twenty Thousand Leagues Under the Sea English-French parallel corpus, 简称 English-20K) 作为跨领域的测试数据。这些分割好的语料能够在 SemEval-2019 Task 1 的官方网站^①上找到，详细数据见表 1。在尝试句法增强的实验中，我们使用英文宾夕法尼亚树库 (English Penn Treebank, 简称 PTB) 作为依存句法的语料。经统计，句法的语料与两个 UCCA 语料没有任何交叉。

评价指标。为了评估两个 UCCA 有向无环图的相似度，我们使用 DAG F1 分数。形式上来讲，给定在同一个句子上的两个 UCCA 图 G_p 和 G_g ，对于其中的任意一个非叶子节点 v ，定义 $yield(v)$ 为 v 的所有叶子子孙。 G_p 中的一条边 $(v_p \rightarrow u_p)$ 与 G_g 中的一条边 $(v_g \rightarrow u_g)$ 匹配当且仅当 $yield(u_p) = yield(u_g)$ 并且它们的标签相同。带标签的准确率和召回率即为两个图中匹配的边数分别除以 G_p 中的边数和 G_g 中的边数，F 分数就是准确率和召回率的调和平均值。我们使用官方提供的脚本来评估主边以及远程边的 F 值，标点符号仅在评估时被排除在外。

表 1 UCCA 以及句法语料数据
Table 1 The statistics of UCCA and dependency corpus

Corpus	Train		Dev		Test		Total	
	Sentences	Tokens	Sentences	Tokens	Sentences	Tokens	Sentences	Tokens
English-Wiki	4113	124935	514	17784	515	15854	5142	158573
English-20K	0	0	0	0	492	12574	492	12574
PTB	39832	950028	1700	40117	2416	56684	43948	1034093

4.1 实验设置

所有的实验都采用了 Adam 优化器(Kingma and Ba, 2014^[19])，学习速率为 0.001。迭代的最大次数设置为 100，当 UCCA 开发集上的平均 F 值经过 15 次迭代后仍然没有提升，则训练提前结束。在使用句法感知微调词向量方法同时训练两个模型时，我们从两个语料中分别选取一个 batch 并更新一次参数。由于句法的训练集远大于 UCCA 的训练集，我们设置句法的 batch size 等于 UCCA 的 batch size 乘上他们训练集之间的相应倍数，确保两个训练集在相同的 batch 数中完全使用。

请注意，由于句法语料中的词性规范和 UCCA 语料中的完全不同，我们在两个基准模型上做了部分修改。为了保证句法模型能接收 UCCA 语料中的句子，我们将两个模型中的自动词性向量都替换成了 Lample et al. (2016)^[20]的字符 BiLSTM。具体来讲就是每个词的所有字输入到正向和反向的 LSTM 中，将它们的输出拼接起来作为字符特征。我们在 PTB 上单独训练了一个 Biaffine Parser，它在测试集上的 LAS (Labeled Attachment Score) 值达到了 94.39%，与 Dozat et al. (2017)^[12]的结果相当，然后利用它生成 UCCA 语料对应的依存句法树或者特征向量作为额外的输入资源。

^① <https://competitions.codalab.org/competitions/19160>

为了进一步探究句法对于语义分析的作用，我们将加入 BERT 后的模型作为更强的基线模型。我们从官方提供的预训练英文 BERT 模型^①中提取固定特征向量作为额外输入，这些基于上下文的词表示来自 BERT 编码器的后四层，像 ELMo 一样经过加权求和再乘上一个特定任务相关的系数，拼接在词表示向量之后输入到上层的编码器。

4.2 实验结果

表 2 中第一主行的两个结果来自 Hershovich et al. (2019)^[21]，显然该基于转移的模型效果不如基于图的分析模型，平均 F 值比本文的基线模型低了 4.4%-5.1%。在利用了其它语义任务后，本领域提升 0.7%，跨领域提升 1.2%，表明了不同语义任务之间的互助作用。

表 2 中的第二主行展示了基线模型以及使用四种句法增强方法后的结果。与 Jiang et al. (2019)^[5]的结果相比，本文的基线模型在开发集上的平均 F 值几乎一样，在两个测试集上分别有 0.2%和 0.4%的下降，说明删除了提供的自动依存标签和实体标签向量影响并不大，而且字符 BiLSTM 一定程度上能取代自动词性向量。使用了四种句法增强的方法后，三个数据集上的平均 F 值都有显著提升（开发集提升 0.7%-1.4%，本领域测试集提升 0.9%-1.6%，跨领域测试集提升 1.1%-1.9%），表明句法能有效的增强 UCCA 语义分析。其中效果最差的是依存标签嵌入方法，因为它仅仅利用了依存树的局部信息，而不像 Tree-GRU 能够编码整棵树。句法感知词表示方法和 Tree-GRU 方法的效果十分接近，我们认为这是因为句法编码器的各层输出本身就蕴含了所有句法的隐式信息，它与句法树的编码信息等价。四种方法中效果最好的是微调句法感知词表示的方法，它在三个数据集上的结果均最高。值得一提的是该模型训练完毕后，句法模型的 LAS 值为 93.00%，比直接预训练的模型低了超过 1%，这表明该方法在利用句法的隐式信息的同时，牺牲了句法的部分性能，强迫句法模型去学习语义知识，从而产生的词表示可能还含有一定的语义信息，更加适用于 UCCA 语义分析任务。

表 2 引入句法的实验结果。
Table 2 Results of experiments utilizing dependency data.

方法	Dev			Wiki Test			20K Test		
	Primary	Remote	Average	Primary	Remote	Average	Primary	Remote	Average
Hershovich et al. (2017)	74.5	53.4	74.1	73.3	47.2	72.8	68.2	23.7	67.2
Hershovich et al. (2018)	75.3	51.4	74.8	73.9	53.5	73.5	69.2	25.9	68.4
Jiang et al. (2019)	79.4	53.5	78.9	77.9	52.2	77.4	73.6	31.2	72.7
Baseline(Our)	79.5	43.8	79.0	77.6	48.0	77.2	73.3	19.0	72.3
Without									
依存标签嵌入	80.2	49.8	79.7	78.6	53.9	78.1	74.4	28.4	73.4
BERT									
Tree-GRU 编码	80.4	52.4	79.9	78.9	54.4	78.5	75.0	30.5	74.0
句法感知词表示	80.5	49.9	80.0	78.7	50.4	78.3	75.0	24.0	73.9
微调句法感知词表示	80.9	54.4	80.4	79.2	58.8	78.8	75.1	32.4	74.2
Jiang et al. (2019)	82.5	60.3	82.1	81.0	58.8	80.5	77.7	39.2	76.7
Baseline(Our)	83.0	57.4	82.6	81.6	59.3	81.2	78.2	35.5	77.2
With									
依存标签嵌入	83.1	54.0	82.6	82.1	56.9	81.6	78.5	31.4	77.5
BERT									
Tree-GRU 编码	83.1	59.5	82.6	81.5	57.6	81.0	77.9	41.1	77.0
句法感知词表示	83.1	58.2	82.6	81.7	58.0	81.3	78.4	38.6	77.5
微调句法感知词表示	82.9	51.8	82.4	81.8	57.1	81.3	78.4	35.8	77.5

表 2 中的第三主行展示了加入 BERT 特征后的结果，其中每一行的模型都加入了 BERT 特征作为额外输入。本文的基线模型明显好于 Jiang et al. (2019)^[5]是因为他们使用了多语言 BERT 而本文使用了单语言 BERT。我们可以看到在 UCCA 基础模型上加入 BERT 特征后平均 F 值有巨大的提升（本领域测试集提升

① 我们使用的 BERT 模型来自 <https://github.com/google-research/bert>。

4%，跨领域测试集提升 4.9%)。以此作为基准线，使用了四种句法增强的方法，除了 Tree-GRU 反而使结果有些下降。其他方法仍然有一些提升(0.1%-0.4%)，最好的方法是直接利用依存标签。可能的原因是 Tree-GRU 这种全局的显示句法信息带来了噪声，与 BERT 模型中的隐式句法信息产生了冲突，而依存标签这种局部的句法信息反而带来了一定的帮助。

5 结论

本次工作中，我们探究了句法对于 UCCA 语义分析的作用。我们提出并尝试了四种不同的方法来利用句法信息，包括基于依存标签嵌入的方法、基于 Tree-GRU 句法树编码的方法、基于句法感知词表示的方法以及基于句法感知微调词表示的方法。实验表明，这些方法都能直接有效地提升 UCCA 语义分析性能，其中基于句法感知微调词表示的效果最好，因为它不仅使用了编码器的隐式句法信息，同时让句法模型倾向于学习语义知识。在引入了 BERT 之后，由于语言模型本身能学习到隐式的句法信息，依存句法模型产生的显示句法信息作用就削减了，但是除了 Tree-GRU 这种全局的编码方法产生了副作用，其他方法也仍然有 0.1%-0.4%的提升，一定程度上表明了依存句法对于 UCCA 语义分析的贡献。

参考文献

- [1] Abend O and Rappoport A. Universal Conceptual Cognitive Annotation(UCCA). In Proceedings of ACL, 2013: 228–238
- [2] Abend O, Yerushalmi S, and Rappoport A. UCCA App: Web-application for syntactic and semantic phrase-based annotation. In Proceedings of ACL: System Demonstration Papers. 2017: 109–114
- [3] Birch A, Abend O, Bojar O, et al. HUME: Human UCCA based evaluation of machine translation. In Proceedings of EMNLP, 2016: 1264–1274
- [4] Hershovich D, Abend O, and Rappoport A. A transition-based directed acyclic graph parser for ucca. In Proceedings of ACL, 2017: 1127–1138
- [5] Jiang Wei, Li Zhenghua, Zhang Yu, et al. HLT@SUDA at SemEval-2019 Task1: UCCA Graph Parsing as Constituent Tree Parsing. arXiv preprint, 2019, arXiv:1903.04153
- [6] Hershovich D, Abend O, and Rappoport A. Multitask parsing across semantic representations. In Proceedings of ACL, 2018: 373–385
- [7] Xia Qingrong, Li Zhenghua, Zhang Min, et al. Syntax-aware Neural Semantic Role Labeling. In Proceedings of AAAI, 2018
- [8] 江腾蛟, 万常选, 刘德喜, 等. 基于语义分析的评价对象-情感词对抽取. 计算机学报, 2017, 40(03): 617–632
- [9] 张志远, 赵越. 基于语义和句法依存特征的评论对象抽取研究. 中文信息学报, 2018, 32(06): 85–92
- [10] Devlin J, Chang Mingwei, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint, 2018, arXiv:1810.04805
- [11] Stern M, Andreas J, Klein and D. A minimal span-based neural constituency parser. In Proceedings of ACL, 2017: 818–827
- [12] Dozat T and Manning C D. Deep biaffine attention for neural dependency parsing. In Proceedings of ICLR, 2017
- [13] Tai Kaisheng, Socher R, and Christopher D. Manning. Improved semantic representations from tree-structured long short term memory networks. In Proceedings of ACL, 2015: 1556–1566
- [14] Miwa M, and Bansal M. End-to-end relation extraction using lstms on sequences and tree structures. In Proceedings of ACL, 2016: 1105–1116
- [15] Feng Yutian, Zhang Hongjun, Hao Wenning, et al. Joint extraction of entities and relations using reinforcement learning and deep learning. In Proceedings of Computational Intelligence and Neuroscience. 2017: 1–11
- [16] Chen Huadong, Huang Shujian, Chiang David, et al. Improved neural machine translation with a syntax-aware encoder and decoder. In Proceedings of ACL, 2017: 1936–1945
- [17] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations. In Proceedings of NAACL. 2018: 2227–2237
- [18] Zhang Meishan, Li Zhenghua, Fu Guohong et al. Syntax-Enhanced Neural Machine Translation with Syntax-Aware Word Representations. In Proceedings of NAACL. 2019: 1151–1161
- [19] Kingma D and Ba J. Adam: A method for stochastic optimization. CoRR, 2014, abs/1412.6980

- [20] Lample G, Ballesteros M, Subramanian S, et al. Neural Architectures for Named Entity Recognition. In Proceedings of NAACL. 2016: 260–270
- [21] Hershcovich H, Aizenbud Z, Choshen L, et al. SemEval-2019 Task1: Cross-lingual Semantic Parsing with UCCA. arXiv preprint, 2019, arXiv:1903.02953