语义结构的句法支架

Swabha Swayamdipta♠ Sam Thomson♠
Kenton Lee♦ Luke Zettlemoyer♦ 克里斯戴尔♥ 诺亚史密斯♦♣

♠语言技术研究所，卡内基梅隆大学，宾夕法尼亚州匹兹堡
♦Google AI Language，Seattle WA，USA
♦Paul G. Allen华盛顿大学计算机科学与工程学院，美国华盛顿州西雅图市
♥Google DeepMind，英国伦敦
♣Allen Institute for Artificial Intelligence，Seattle，WA，USA
{swabha，sammthomson} @ cs.cmu.edu {kentonl，cdyer} @ google.com
{LSZ，nasmith} @ cs.washington.edu

§

摘要

我们介绍了句法支架，一种将语法信息结合到语义任务中的方法。 句法脚手架在运行时避免了昂贵的句法处理，只在训练期间通过多任务目标使用树库。 我们改进了PropBank语义，框架语义和共指消解的强大基线，在所有三个任务上实现了竞争性能。

1 介绍

随着自然语言句子语义分析算法的发展，语法的作用不断被重新审视。 语言学理论认为句法和语义处理的紧密结合（骏马- 人，2000; Copestake和Flickinger，2000），许多系统使用语法依赖或基于短语的解析器作为语义分析的预处理（吉尔德和帕尔默，2002; 双关语- yakanok等。，2008; 达斯等人。，2014）。同时，最近的一些方法完全禁止显式语法处理（周和徐，2015; 他 等。，2017; 李等人。，2017; 彭等人。，2017）.

因为用于语义的带注释的训练数据集将始终受到限制，我们期望语法 - 其提供不完整但可能有用的语义结构视图 - 将继续提供有用的归纳偏差，从而鼓励语义模型朝向更好的泛化。 我们解决了一个核心问题：语法分析器是否有一种方法可以从语法中受益而无需语法分析的计算成本？

我们提出了一种多任务学习方法，将句法信息纳入学习中神经语义模型的表示（ 2）。 我们的方法，句法支架，最小化从句法树库派生的辅助监督损失函数。 目标是引导单词和跨度的分布式语境化表示朝向准确的语义和句法标签。我们避免了培训或执行完整语法分析器的成本，并且在测试时（即应用程序中的运行时），语义分析器在无语法基线上没有额外成本。 此外，该方法不假设语法树库与主要任务的数据集重叠。

许多语义任务涉及标记跨度，包括语义角色标记（SRL; 吉尔德和 Jurafsky，2002）和共同决议（伍，2010）（我们在本文中考虑的任务），以及命名实体识别和一些阅读理解和问答环节（岭 jpurkar等。，2016）。 这些跨度通常是句法成分（参见PropBank; 帕尔默等。，2005），使基于短语的语法成为脚手架的自然选择。 见图 1 对于具有句法和语义注释的示例句子。 由于脚手架任务本身并不是目的，我们将语法分析问题放松到独立跨度级预测的集合，没有约束它们形成有效的分析树。 这意味着我们永远不需要运行语法分析算法。

我们的实验表明，语法支架为两个SRL任务的基线状态提供了实质性的推动力（ 5）和共同决议（ 6）。 我们的模型使用最强大的可用神经网络架构来完成这些任务，集成深度表示学习（他 等。，2017）和跨度水平的结构化预测（Kong等。，2016）。 对于SRL，基地 -

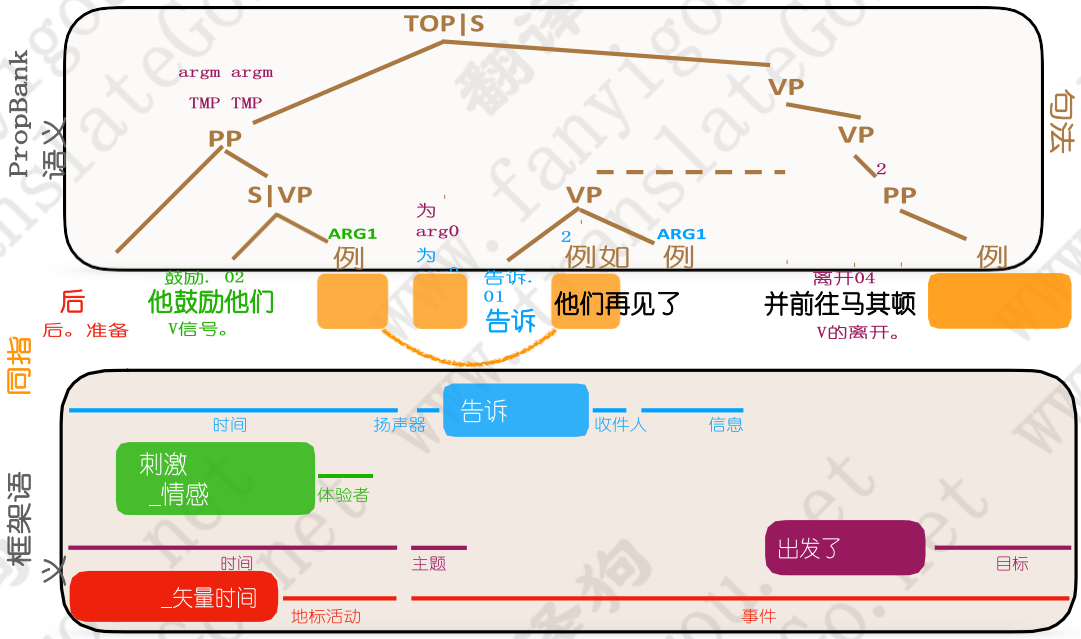

图1：来自OntoNotes的语法，PropBank和共同参考注释以及作者注释的框架语义结构的示例句子。 PropBank SRL参数和共参考提及在句法成分之上注释。 除了一个帧语义参数（Event）之外的所有参数都是语法成分。 目标唤起颜色编码层中显示的帧。

线本身是一种新颖的全局归一化结构化条件随机场，其优于先前的技术水平。[1] 句法脚手架比以前的工作进一步改善 - 3.6 FrameNet SRL中的绝对F_1，PropBank SRL中的1.1绝对F_1和共参考分辨率中的0.6 F_1（三个标准分数的平均值）。 我们的代码是开源的，可在以下处获得 HTTPS：支架式swabhs github.com / / / /.

2 句法脚手架

多任务学习（卡鲁阿纳，1997）是一组技术，其中从至少一些参数共享的数据中学习两个或更多个任务。 我们假设我们关注的性能只有一个任务，表示为T_1（在本文中，T_1是SRL或共同参照分辨率）。 我们使用术语“支架”来指代第二个任务T_2，其可以在多任务学习期间与T_1组合。 脚手架任务仅在训练期间使用；除了偏向T_1的学习之外，它没有内在的兴趣，并且在完成学习之后，丢弃了支架。

句法支架是一项旨在引导（共享）模型转向语法意识的任务结构体。 它可以通过一个语法分析器来定义，该分析器与T_1的模型共享一些参数。 由于语法分析代价高昂，我们使用更简单的句法预测问题（下面讨论），不会产生整个树。

与一般的多任务学习一样，我们不假设相同的数据用T_1和T_2的输出进行注释。 在这项工作中，T_2是使用OntoNotes 5.0中的短语结构语法注释定义的（Weischedel等。，2013；普拉丹 等。，2013）。 我们尝试了三种设置：一种是T_2的语料库与T_1（frame-SRL）的训练数据集不重叠，另一种是完全重叠（PropBank SRL和共同参考）。 与在相同数据上需要多个输出标签的方法相比，我们提供的主要优点是不需要对T_1和T_2输出之间的关系进行任何假设或规范。

3 相关工作

我们简要地将句法支架与现有的替代方案进行对比。

管道。 在典型的流水线中，T_1和T_2分别经过训练，T_2的输出用于定义T_1的输入（沃伯特，1992）。 在管道中使用语法T_2可能是最多的

[1]这排除了使用深度语境化嵌入初始化的模型（彼得斯等人。，2018），一种与我们正交的方法。

语义结构预测的常用方法（Toutanova等。，2008; 杨和米切尔，2017; Wiseman等人。，2016).[2] 然而，管道引入了级联错误的问题（T_2的错误会影响T_1的性能，也许会影响训练。 他等人。，2013）。 到目前为止，对级联错误的补救措施在计算上非常昂贵而且不切实际（例如， 芬克尔等人。，2006）。 句法支架与管道完全不同，因为从未明确使用T_2的输出。

潜在变量。 另一种解决方案是将T_2的输出视为（可能是结构化的）潜在变量。 这种方法不需要监督T_2，并且需要边缘化（或对它的一些近似）以推断T_1的输出。 语法作为语义的潜在变量进行了探索 泽特尔莫耶 和柯林斯(2005）和 Naradowsky等人。 (2012）。 除了避免边缘化之外，句法支架提供了一种使用辅助语法注释数据作为T_2的直接监督的方法，并且它不需要与T_1训练数据重叠。

联合学习语法和语义。 联合学习句法和语义表达背后的动机是任何一项任务都有助于预测另一项任务（Llu'ıs和Ma`rquez，2008; Llu'ıs等。，2013; 亨德森等人。，2013; Swayamdipta等。，2016）。 这通常需要联合预测T_1和T_2的输出，这在训练和测试时间往往是计算上昂贵的。

部分演讲脚手架。 与我们的工作类似，已经有多任务模型使用部分语音标记作为T_2，基于转换的依赖性解析（张和韦斯，2016）和CCG supertagging （Søgaard和Goldberg，2016）作为T_1。 上述两种方法均假设并行输入数据，并将这两项任务用作监督。 值得注意的是，我们简化了T_2，抛弃了语法分析的结构化方面，而词性标注的结构却很少。 虽然他们的方法导致通过POS标签监督学习的令牌级表示得到改善，但这些表示仍必须如此

组合以获得跨度表示。 In-

[2]最近有一些关于SRL的工作

完全放弃句法处理（周和徐，2015但是，已经表明，合并句法信息仍然有用（他等人。，2017).

对于语义任务，我们的方法直接从短语类型监督中学习跨度级表示。 此外，这些方法探索RNN层中的架构变化以包括监督，而我们专注于将监督与基线架构的最小变化结合起来。据我们所知，这种简化的句法支架以前没有尝试过。

Word嵌入。 我们对支架任务的定义几乎包括估算字嵌入的独立方法（Mikolov等。，2013; Pennington等。，2014; 彼得斯等人。，2018）。在训练单词嵌入之后，像skip-gram或ELMo的语言模型这样的模型隐含的任务变得与嵌入的下游使用无关。 一个值得注意的区别是，通过多任务目标，将支架直接集成到T_1的训练中，而不是预训练。

多任务学习。 当一起训练多个任务时，神经架构通常会产生性能提升（Collobert等。，2011; 鲁- ong等。，2015; 陈等人。，2017; 桥本等。，2017）。 特别是，当与其他语义任务一起完成时，语义角色标记任务的性能得到改善（菲茨杰拉德等人。，2015; 彭等人。，2017，2018）。 与这项工作同时进行，Hershcovich等人。 (2018）提出了一个针对通用语法依赖和UCCA语义的多任务学习设置（Abend和 拉波波特，2013）。 句法脚手架专注于主要语义任务，将语法视为辅助，最终忘记预测任务。

4 句法脚手架模型

我们假设两个监督源：一个语料库$_1$，其实例x注释主要任务的输出y（语义角色标记或共指消解），以及一个带有句子x的树库$_2$，每个句子都有一个短语结构树z。 D

4.1损失

每项任务都有相关损失，我们力求最大限度地减少任务损失的组合，

$$\sum_{(x,\,y)\in D_1} L_1(x,\,y) + \delta \sum_{(X,\,Z)\in D_2} L_2(x,\,z) \quad (1)$$

关于部分共享的参数，其中δ是可调超参数。在

在本节的其余部分，我们描述了脚手架任务。 我们在Sections中定义主要任务 5–6。

每个输入都是一系列令牌，$x = (x_1, x_2, \ldots, x_n)$，对于某些n。 我们将句子中的一系列连续令牌称为$x_{i:j} = (x_i, x_{i+1}, \ldots, x_j)$，任何1 i j n。 在我们的实验中，我们认为只能跨越最大值妈妈长度D，导致O（nD）跨度。

监督来自句子的短语 - 句法树z，包括x中每个跨度$x_{i:j}$的句法类别$z_{i:j}$（许多跨度被赋予空标签∅）。 我们尝试不同的标签集（ 4.2).

在我们的模型中，每个跨度$x_{i:j}$由嵌入向量$v_{i:j}$表示（详见§ 5.3）。 分配给$z_{i:j}$的类别的分布来自$v_{i:j}$：

$$p(z_{i:j} = c \mid x_{i:j}) = \underset{c}{\text{softmax}}\ w_c \cdot v_{i:j} \quad (2)$$

其中w_c是与类别c相关的参数向量。 我们将句子中所有跨度的对数损失项加起来给出它的损失：

$$L_2(x, z) = -\frac{1}{D} \sum_{i\ j-i} \log p(z_{i:j} \mid x_{i:j}) 。 \quad (3)$$

4.2 句法脚手架任务的标签

不同类型的句法标签可用于学习语法感知跨度表示：

- **组成身份：** $C = \{0, 1\}$是一个跨度a成分，或不？
- **非终端：** c是跨度的类别，包括非成分的空值。
- **非终端和父级：** c是跨度的类别，与其直接祖先的类别连接在一起。 null用于非成分，用于空祖先。
- **常见的非终端：** 由于大多数语义参数和实体提及都标有少量语法类别，[3] 我们在（i）名词短语（或介词短语，框架SRL）中进行三向分类试验； （ii）任何其他类别； （iii）无效。

在图中 1，对于“鼓励他们”的跨度，组成标识支架标签为1，非终端标签为S VP，非终端标签和父标签为S VP + par = PP，公共非终端标签设置为OTHER。

[3]在OntoNotes语料库中，包括句法和语义注释，44%的语义参数是名词短语，13%是介词短语。

5 语义角色标签

我们提供了一种新的SRL模型，为复合支架的实验提供了强大的基线。 该基线的性能本身与最先进的方法相比具有竞争力（§7).

框架网络。 在FrameNet词典中（面包师傅等。，1998），框架表示一种事件，情境或关系，并与一组语义角色相关联，称为框架元素。 框架可以通过句子中的单词或短语来唤起，称为目标。 然后可以在句子中将诱发帧的每个帧元素实现为句子跨度，称为参数（或者它可以是未实现的）。 给定帧的参数不重叠。

PropBank。 PropBank同样消除了谓词的歧义并识别了参数跨度。 目标被消除歧义为词汇特定的感觉而不是共享框架和一组通用

角色用于所有目标，将参数标签空间减少17倍。最重要的是，参数在语法之上注释

成分，直接耦合语法和语义。 图中提供了两种形式的详细示例 1.

语义结构预测是识别目标，标记其框架或感官以及在句子中标记其所有参数跨度的任务。 这里我们假设黄金目标和框架，并仅考虑SRL任务。

形式上，参数识别的单个输入实例包括：n字句$x = (x_1, x_2, \ldots, x_n)$，单个目标跨度$t = (t_{开始}, t_{结束})$及其诱发帧或感测f。（参数标记）任务是产生句子的分段：$s = (s_1, s_2, \ldots, s_m)$为每个输入x。 段$s = (i, j, y_{i:j})$对应于句子的标记跨度，其中标签$y_{i:j}$ null是跨度填充的角色，如果是，则为null（跨度不会填补任何角色。 在PropBank的情况下，$_f$包含所有可能的角色，$_c \in Y(f)$。对分段进行约束，使得参数跨越覆盖句子并且不重叠（对于s_k，$i_{k+1} = 1 + j_k$；$i_1 = 1$；$j_m = n$）。 允许长度为1的段，使得$i = j$。 针对句子中的每个目标注释预测单独的分段。

5.1 半马尔可夫CRF

为了模拟给定目标的非重叠参数，我们使用半马尔可夫条件随机场（半CRF；Sarawagi等。，2004）。 半CRF在输入序列的标记分段上定义条件分布，并且全局归一化。 通过在参数之间提供保留的空标签，可以将单个tars参数整齐地编码为标记的分段。 半马尔可夫模els比BIO标记方案更强大，已成功用于PropBank

SRL（Collobert等。，2011; 周和徐，2015除其他外，因为半马尔可夫假设允许对（n 1）阶Markov假设下的可变长度段而不是固定长度标签n-gram进行评分。 可以使用半-CRF计算边际可能性

在O（n^2）时间内使用动态编程（ 5.2）。 §通过滤除长于D令牌的段，将其减少到O（nD）。

给定输入x，半CRF定义了一个条件tional distribution p（s | x）。 每个段s（ i，j，$y_{i:j}$被赋予实值得分，ψ（（i，j，$y_{i:j}$）= r，$x_{i:j}$）= w（TF495）$v_{i:j}$，其中$v_{i:j}$是跨度的嵌入（§5.3）和w_r是对应于其标签的参数向量。 得分轮胎分割s是分数的总和其区段：$\Psi(x, s) = \sum_{k=1}^{m} \psi(s_k, x_{i_k:j_k})$。 这些

对分数进行取幂和归一化以定义概率分布。半马尔可夫动态规划算法的和积变体用于计算归一化项（学习期间所需）。 在测试时，max-product变量返回最可能的分段，$s = \arg\max_s \Psi(s, x)$。

学习半CRF的参数以最大化与训练语料库中的金标准段的条件对数似然相关的标准（ 5.2）。 学习者评估和调整句子中每个跨度的片段得分ψ§（s_k，x），这反过来又涉及学习所有跨度的嵌入表示（§5.3).

5.2 Softmax-Margin目标

通常，训练CRF和半CRF模型以最大化条件对数似然目标。 在早期的实验中，我们发现纳入结构化成本是有益的； 我们通过使用softmax-margin培训目标来实现这一目标（金佩尔和史密斯，2010），“成本意识”变体对数似然：

$$L_1 = -\sum_{(X, S^*) \in D_1} \log \frac{\exp \Psi(s^*, x)}{Z(x, s^*)}, \quad (4)$$

$$Z(x, s^*) = \sum_{s} \exp\{\Psi(s, x) + \text{cost}(s, s^*)\}。 \quad (5)$$

我们设计成本函数，使其以预测跨度为因子，与Ψ的方式相同：

$$\text{成本}(s, s^*) = \sum_{\text{是的}\in} \text{成本}(s, s^*) = \sum_{\text{是的}\in} I(s < 1\ s^*)。 \quad (6)$$

softmax-边际标准，如对数似然，在所有指数多个可能的标记分段上全局归一化。 以下的零阶半马尔可夫动态程序（Sarawagi等。，2004）有效地计算新的分区函数：

$$\alpha_j = \sum_{s=(i, j, y_{i:j})} \alpha_{i-1} \exp\{\Psi(s, x) + \text{cost}(s, s^*)\}, \quad (7)$$
$$j-i\ D$$

其中Z =α_n，在基本情况下α_0= 1。

模型下的预测可以使用类似的动态程序计算，具有以下重现，其中γ_0= 1：

$$\gamma_j = \max_{s, y_{i:j}} \gamma_{i-1} \exp \Psi(s, x)。 \quad (8)$$
$$J-\text{的}iD$$

我们的模型公式强制要求参数不重叠。 我们不强制执行任何其他SRL约束，例如不重复核心框架元素（达斯等人。，2012).

5.3 输入跨度表示

本节描述用于获得跨度嵌入的神经结构$v_{i:j}$，对应于跨度$x_{i:j}$和考虑的目标，t = $t_{\text{开始}}$，$t_{\text{结束}}$。对于scaffold任务，由于语法树库不包含语义目标的注释，因此我们使用句子中的最后一个动词作为占位符目标，无论使用何种目标特征。 如果没有动词，我们使用句子中的第一个标记作为占位符目标。 用于学习v的参数在任务之间共享。

我们使用构建跨度的嵌入

- h_i和h_j：跨度边界处的单词的语境化嵌入（§5.3.1），
- $u_{i:j}$：汇总跨度内容的跨度摘要（§

5.3.2），和

- $\mathbf{a}_{i:j}$：和跨度的手工设计特征向量（§ 5.3.3）。

然后将该嵌入传递到前馈层以计算跨度表示$\mathbf{v}_{i:j}$。

5.3.1 语境化令牌嵌入

为了获得输入序列中每个标记的上下文嵌入，我们运行双向LSTM（格雷夫斯，2012）在整个输入序列上有e层。 为了指示哪个令牌是谓词，使用线性变换的单热嵌入v，如下 周和徐 (2015) 和 他 等。 (2017)。 表示句子中位置q处的标记的输入向量是固定预训练嵌入$\mathbf{x}_q$和$\mathbf{v}_q$的串联。 当作为双向LSTM的输入给出时，这产生表示句子上下文中的第q个标记的隐藏状态向量$\mathbf{h}_q$。

5.3.2 跨度摘要

跨度内的标记可以传达将跨度标记为语义参数所需的不同数量的信息。 以下 李等人。(2017)，我们使用注意机制（Bahdanau 等。，2014）总结每个跨度。 跨度中的每个上下文标记都通过前馈网络以获得权重，归一化以给出$\sigma_k = \text{softmax}(\mathbf{w}_{\text{头}} \cdot \mathbf{h}_k)$，其中$\mathbf{w}_{\text{头}}$是一个学习参数。 然后是权重$\sigma$用于获得总结跨度的向量，$\mathbf{u}_{i:j} = \sum_{k=i}^{j} \sigma_k \cdot \mathbf{h}_k$。

5.3.3 跨度特征

我们为每个跨度使用以下三个功能：

- 标记中的跨度宽度（达斯等人。，2014）
- 距离焦距的距离（标记）

  得到（Täckström 等。，2015）
- 跨度相对于目标的位置（之前，之后，重叠）（Täckström 等。，2015）

将这些特征中的每一个编码为单热嵌入，然后线性变换以产生特征向量，即$\mathbf{a}_{i:j}$。

6 共同决议

核心参考解决方案是确定引用同一实体的提及集群的任务。 形式上，输入是文档$x = x_1, x_2, \ldots, x_n$由n个单词组成。 目标是预测一组聚类$c = \{c_1, c_2, \ldots\}$。 每个群集$c = \{s_1, s_2, \ldots\}$是一组跨度和

每个跨度$s = (i, j)$是一对索引，使得$1 \le i \le j \le n$。

作为基线，我们使用的模型 李等人。(2017)，我们在本节中简要介绍。 该模型将共参照簇的预测分解为一系列跨度分类决策。每个跨度都预示着一个先行者$w_s \in Y(s) = \{\text{null}, s_1, s_2, \ldots, s_m\}$。 标签$s_1$到$s_m$表示s与其前面的m个跨度之一之间的共参考链接，null表示s不会链接到任何东西，要么是因为它不是提及，要么是在单个群集中。 可以通过聚合预测的链接来恢复预测的跨距聚类。

类似于SRL型号（ 5），每个跨度s由嵌入$\mathbf{v}_s$表示，嵌入v是模型的核心。 对于每个跨度s和潜在的先行词a（s），成对共参数得分$\Psi(\mathbf{v}_s, \mathbf{v}_a, \varphi(s, a)) \in Y$通过前馈网络以跨度嵌入作为输入来计算。 $\varphi(s, a)$是编码跨度s和跨度a与元数据之间的距离的成对离散特征，例如类型和说话者信息。 我们推荐读者 李等人。 (2017)有关评分功能的详细信息。

来自Ψ的分数在每个跨度的可能前因（s）上进行归一化，以得出每个跨度的概率分布：

$$p(w_s = a) = \text{softmax}_{a \in Y(s)}(\Psi(\mathbf{v}_s, \mathbf{v}_a, \varphi(s, a))) \quad (9)$$

在学习中，我们最大限度地减少了可能正确的前因而被边缘化的负对数似然：

$$L_1 = -\log \prod_{\text{的}\in D} \sum_{\text{一个}^* \in G(s) \cap Y(s)} p(w_s = a^*) \quad (10)$$

其中是训练数据集中的跨距集合，G（s）表示s的黄金集群（如果它属于一个，否则为null）。 { }

为了在合理的计算要求下操作，在该模型下的推断需要两阶段波束搜索，这减少了所考虑的跨度对的数量。 我们推荐读者 李等人。(2017) 详情。

输入范围表示。 用于共参分辨率的输入跨度嵌入，$\mathbf{v}_s$及其句法支架遵循用于的定义 5.3，与使用无目标功能的关键区别。 由于SC和公关之间的输入句子完全重叠，因为共同注释也来自OntoNotes（Pradhan等人。，

2012），我们将v重用于脚手架任务。 另外，代替整个文档，其中的每个句子独立地作为双向LSTM的输入给出。

7 结果

我们在FrameNet 1.5的测试集上为框架SRL评估我们的模型，在OntoNotes的测试集上评估我们的模型，用于PropBank SRL和共同参考。 对于每种情况下的句法支架，我们使用OntoNotes的句法注释
5.0 (Weischedel等。，2013; Pradhan等人。，2013).[4] 有关实验设置和数据集的更多详细信息已在补充材料中详细说明。

Frame SRL。 表 1 显示了框架SRL上所有支架模型相对于先前工作和半CRF基线的性能（ 5.1）没有句法支架。 我们遵循SemEval共享任务的官方评估进行框架语义分析（贝克等人。，2007).

框架SRL的先前工作依赖于预先
以两种不同的方式编写句法树：通过使用基于语法的规则来删除不太可能包含任何框架参数的文本跨度； 并在统计模型中使用句法特征（达斯等人。，2014; Täckström等。，2015; 菲茨杰拉德等人。，2015; Kshirsagar等。，2015).

在FrameNet 1.5上发布的最佳结果是由于杨和米切尔 (2017）。 在他们的序列模型（seq）中，他们使用具有CRF层的深双向LSTM将参数识别视为序列标记问题。 在他们的关系模型（Rel）中，他们将相同的问题视为跨度分类问题。 最后，他们介绍 -

使用一个整体来整合两个模型，和
使用整数线性程序进行推理以满足SRL约束。尽管他们的模型没有进行任何语法修剪，但它确实使用语法特征进行参数识别和标记。[5] 值得注意的是，表中列出了所有现有的框架SRL系统 1 使用语法和语义管道。 我们的半CRF基线优于所有先前的工作，没有任何语法。

这凸显了本 -

[4] http://cemantix.org/数据/ontonotes.html

[5] 杨和米切尔 (2017）还评估了完整的帧语义分析任务，其中包括frame-SRL as

以及识别框架。 由于我们的框架SRL性能
改进了它们，我们希望将其整合到一个完整的系统中（例如，使用它们的框架识别模块）也可以带来整体效益；这个实验留待将来的工作。

建模跨度和全局规范化的效果。

转向支架，即使是最粗糙的组成身份支架也可以改善我们的语法无关基线的性能。 使用更详细的句法表示的非终结和非终结和父母支架改进了这一点。 最大的改进来自支架模型预测常见的非终结标签（NP和PP，这是语义论证最常见的句法类别，与其他语法论证相比）：F_1衡量的先前工作的绝对改进率为3.6%。

与这项工作同时进行， 彭等人。 (2018）提出了一种用于联合框架语义和语义依赖性解析的系统。 它们报告了关节框架和参数识别的结果，因此无法在表格中直接进行比较 1。 我们仅评估其输出的参数识别；我们的半CRF基线模型超过其性能1 F_1，我们常见的非终端支架3.1 F_1。[6]

模型	PREC。	建议	F_1
Kshirsagar等。 (2015)	66.0	60.4	63.1
杨和米切尔 (2017） (Rel)	71.8	57.7	64.0
杨和米切尔 (2017） (Seq)	63.4	66.4	64.9
†杨和米切尔 (2017） (全部)	70.2	60.2	65.5
半CRF基线	67.8	66.2	67.0
+组成身份	68.1	67.4	67.7
+非终结者和父母	68.8	68.2	68.5
+非终结	69.4	68.0	68.7
+常见的非终结者	69.2	69.0	**69.1**

表1：Frame SRL在FrameNet 1.5。的测试集上的结果，使用金框。 合奏用†表示。

模型	PREC。	建议	F_1
周和徐 (2015)	-	-	81.3
他等人。 (2017)	81.7	81.6	81.7
他等人。 (2018a)	83.9	73.7	82.1
Tan等人。 (2018)	81.9	83.6	82.7
半CRF基线	84.8	81.2	83.0
+常见的非终结者	85.1	82.6	**83.8**

表2：使用黄金谓词的PropBank sSRL结果，在CoNLL 2012测试中。 为了公平比较，我们只展示非整体模型。

[6] 该结果未在表中报告 1 以来 彭等人。 (2018）使用预处理使测试集略大 - 我们报告的差异是使用他们的测试集计算的。

模型	MUC			B^3			$CEAF_{\varphi 4}$			平均。
	PREC。	建议	F_1	PREC。	建议	F_1	PREC。	建议	F_1	
Wiseman等人。 (2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
克拉克和曼宁 (2016b)	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
克拉克和曼宁 (2016a)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
李等人。 (2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
+常见的非终结者	78.4	74.3	76.3	68.7	62.9	65.7	62.9	60.2	61.5	**67.8**

表3：英语CoNLL-2012共享任务上的测试集上的Coreference解析结果。 MUC，B^3和$CEAF_{\varphi 4}$的平均F_1是主要评估指标。 为了公平比较，我们只展示非整体模型。

PropBank SRL。 我们在2012年使用CoNLL共享任务中的OntoNotes数据（普拉丹 等。，2013）对于Propbank SRL。 表 2 使用黄金谓词报告结果。

PropBank SRL最近的竞争系统遵循以下方法 周和徐 (2015)，采用深层体系结构，并放弃使用任何语法。 他等人。 (2017）改进这些结果，并在分析实验中，表明使用语法导出的约束可以进一步提高性能。 Tan等人。(2018采用类似的方法，但使用具有自我关注的前馈网络。 他等人。 (2018a）使用基于跨度的分类来共同识别和标记参数跨度。

我们的语法无关的半CRF基线模型改进了先前的工作（不包括ELMo），再次显示了语义结构预测中全局规范化的价值。 我们用来自框架SRL任务的最佳句法支架获得0.8绝对F_1的进一步改进。 这表明即使使用复杂的神经架构，句法归纳偏差也是有益的。

他等人。 (2018a）还提供了一个设置，其中初始化使用深度上下文嵌入，ELMo（彼得斯等人。，2018）， 导致OntoNotes测试集上的85.5 F_1。 ELMo的改进在方法上与句法支架正交。

由于用于学习PropBank语义和句法支架的数据集完全重叠，因此性能改进不能归因于更大的训练语料库（或者，通过扩展，更大的词汇表），尽管这可能是帧SRL的一个因素。

句法支架可以匹配包含精心提取的句法特征的管道的性能，用于语义预测（Swayamdipta 等。，2017）。 这与其他近期的方法（他等人。，2017，2018b）表明即使对于SRL的强神经模型，语法仍然有用。

同指。 我们报告了CoNLL评估的四个标准分数的结果：MUC，B^3和$CEAF_{\varphi 4}$，以及它们在表中的平均F_1 3。 先前的竞争性共指解析系统（明智的- 男人等。，2016；克拉克和曼宁，2016b,a）所有都包含管道中的同义信息，使用来自预测语法的提议提议的特征和规则。

我们的基线是来自的模型 李等人。(2017)，描述于 6§ 与帧SRL的基线模型类似，与之前的工作相比，此模型不使用任何语法。

我们从框架SRL任务中试验最好的句法支架。我们在这里使用NP，OTHER和null作为常见非终端支架的标签，因为coreferring提及很少是介词短语。 句法支架的表现优于基线0.6绝对F_1。同时， 李等人。 (2018）提出了一个模型，该模型考虑了更高阶的推断和更积极的修剪，以及使用ELMo嵌入进行初始化，得到73.0的平均F_1。 以上所有内容都与我们的方法正交，并且可以合并以产生更高的收益。

8 讨论

为了研究句法支架的性能，我们关注框架SRL结果，其中我们观察到非语法基线的最大改进。

我们考虑通过FrameNet中提供的参数的语法短语类型对性能进行细分[7]在图中 2。 不奇怪 -

[7]我们使用FrameNet语法短语注释仅用于分析，而不是在我们的模型中，因为它们仅针对黄金参数进行注释。

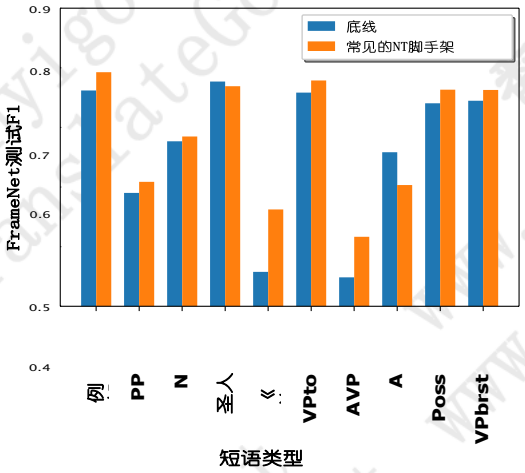

图2：按照argus短语类别的性能细分，按频率从左到右排序，排名前十位。

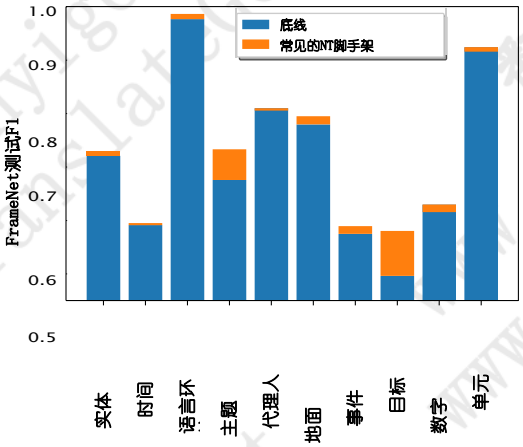

图3：前十个框架元素类型的性能细分，按频率从左到右排序。

我们观察到使用的常见非终端（NP和PP）有很大改进。 但是，FrameNet中的短语类型注释与OntoNotes短语类别并不完全对应。 例如，FrameNet注释非最大（A）和标准形容词短语（AJP），而名词短语的OntoNotes注释是平的，忽略了潜在的形容词短语。 这解释了为什么语法不可知基线能够恢复前者而脚手架不能。

类似地，对于频繁的框架元素，脚手架提高了整个板的性能，如图1所示。 3。 主题和目标最大的改进，主要是作为名词短语和介词短语实现。

9 结论

我们引入了句法支架，这是一种将语法偏差纳入语义处理任务的多任务学习方法。 与管道和语法和语义联合建模的方法不同，运行时不需要显式语法处理。 我们的方法提高了FrameNet和PropBank上语义角色标记的竞争基线的性能，以及共同参照解决方案。 虽然我们的重点是基于跨度的任务，但语法支架可以应用于其他设置（例如，依赖和图形表示）。 而且，支架不需要语法；我们可以想象，例如，语义支架被用于改进具有有限注释数据的NLP应用程序。它仍然是一个开放的经验问题，以确定不同类型的支架和多任务学习者的相对优点，以及它们如何成为最具生产力的结合起来。 我们的代码是公开的http：/ / / / github.com swabhs支架式.

致谢

我们感谢UW-NLP的几位成员，特别是Luheng He，以及David Weiss和Emily Pitler对本文先前版本的深思熟虑的讨论。 我们还要感谢三位匿名审稿人的宝贵意见。 这项工作部分由NSF资助IIS-1562364和NVIDIA公司通过捐赠Tesla GPU提供支持。

参考

Omri Abend和Ari Rappoport。 2013.通用概念认知注释（UCCA）。 在ACL中。

Dzmitry Bahdanau，Kyunghyun Cho和Yoshua Bengio。 通过联合学习对齐和翻译的神经机器翻译。 的arXiv：1409.0473。

Collin Baker，Michael Ellsworth和Katrin Erk。2007. SemEval'07任务19：框架语义结构提取。在Proc。 SemEval。

Collin F. Baker，Charles J. Fillmore和John B. Lowe。 1998.伯克利FrameNet项目。 在Proc。 ACL。

Rich Caruana。 1997.多任务学习。 机器学习，28（1）。

陈新驰，詹士，邱沛鹏，黄玄菁。 2017.中国分词的对抗性多标准学习。 的arXiv：1704.07556。

凯文克拉克和克里斯托弗D曼宁。 2016a。 提升排名共指模型的深度强化学习。 在Proc。 EMNLP。

凯文克拉克和克里斯托弗D.曼宁。 2016B。 通过学习实体级分布式表示来提高共同参考解决方案。 在Proc。 ACL。

Ronan Collobert, Jason Weston, Le'on Bottou, Michael Karlen, Koray Kavukcuoglu和Pavel Kuksa。 2011.自然语言处理（几乎）从头开始。机器学习研究杂志，12：2493-2537。

Ann Copestake和Dan Flickinger。 2000.一个开源语法开发环境和使用HPSG的广泛覆盖的英语语法。 在Proc。 LREC。

Dipanjan Das, Desai Chen, Andre'FT Martins, Nathan Schneider和Noah A Smith。 2014.框架语义解析。 计算语言学，40（1）：9-56。

Dipanjan Das，Andre'FT Martins和Noah A. Smith。 具有约束的浅层语义分析的精确对偶分解算法。 在Proc。 of * SEM。

Jenny Rose Finkel, Christopher D Manning和Andrew Y Ng。 2006.解决级联错误的问题：语言注释管道的近似贝叶斯推断。 在Proc。 EMNLP。

Nicholas FitzGerald, OscarTa¨ckstrom, Kuzman Ganchev和Dipanjan Das。 2015.用神经网络因素标记语义角色。 在Proc。 EMNLP。

Daniel Gildea和Daniel Jurafsky。 2002.自动标记语义角色。 计算语言学，28（3）：245-288。

Daniel Gildea和Martha Palmer。 解析谓词参数识别的必要性。 在Proc。 ACL。

Kevin Gimpel和Noah A. Smith。 2010. Softmax边际CRF：训练具有成本函数的对数线性模型。在Proc。 NAACL。

Alex Graves。 2012.带有回归神经网络的监督序列标记，计算智能研究的第385卷。 斯普林格。

Alex Graves。 2013.用递归神经网络生成序列。的arXiv：1308.0850。

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka和Richard Socher。 2017.一个联合的多任务模型：为多个NLP任务发展神经网络。在Proc。 EMNLP。

何他，Hal Daume'III和Jason Eisner。 2013.依赖性解析的动态特征选择。 在Proc。 EMNLP。

Luheng He，Kenton Lee，Omer Levy和Luke Zettle-moyer。 2018A。 联合预测神经语义角色标注中的谓词和论据。 在Proc。 ACL。

Luheng He，Kenton Lee，Mike Lewis和Luke Zettle-moyer。 2017.深层语义角色标注：什么有效，什么是下一步。 在Proc。 ACL。

何夏霞，李祖超，赵海，白晓霄。 2018B。 语义角色标记的语法，是或不是。 在Proc。 ACL。

James Henderson, Paola Merlo, Ivan Titov和Gabriele Musillo。 2013.使用潜变量模型对语法和语义依赖关系进行多语言联合解析。 计算语言学，39（4）：949-998。

Daniel Hershcovich，Omri Abend和Ari Rappoport。 2018.跨语义表示的多任务解析。在Proc。 ACL。

Diederik P. Kingma和Jimmy Ba。 ADAM：随机优化的一种方法。 的arXiv：1412.6980。

Lingpeng Kong, Chris Dyer和Noah A. Smith。 2016.分段递归神经网络。 在Proc。 ICLR。

Meghana Kshirsagar，Sam Thomson，Nathan Schneider, Jaime Carbonell, Noah A Smith和Chris Dyer。 2015.使用异构注释的帧语义角色标记。 在Proc。 NAACL。

Kenton Lee，Luheng He，Mike Lewis和Luke Zettle-moyer。 2017.端到端神经共指解析。在Proc。 EMNLP。

Kenton Lee, Luheng He和Luke Zettlemoyer。 2018.具有粗tofine推断的高阶共指消解。 在Proc。 NAACL。

XavierLluıs, Xavier Carreras和Llu'ısMa`rquez。 2013.语法和语义依赖关系的联合弧分析解析。ACL的交易，1：219-230。

XavierLlu'ıs和LluısMa`rquez。 2008.用于解析语法和语义依赖关系的联合模型。 在Proc。 CoNLL。

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals和Lukasz Kaiser。 2015.多任务序列到序列学习。 的arXiv：1511.06114。

Tomas Mikolov, Kai Chen, Gregory S. Corrado和Jeffrey Dean。 2013.向量空间中词表示的有效估计。 的arXiv：1301.3781。

Vinod Nair和Geoffrey E. Hinton。 整流线性单元改进了受限制的Boltzmann机器。 在Proc。 ICML。

Jason Naradowsky, Sebastian Riedel和David A. Smith。 通过隐藏的句法结构的边缘化来改善NLP。 在Proc。 EMNLP。

文森特吴 监督名词短语共指研究：前十五年。 在Proc。 ACL。

Martha Palmer, Daniel Gildea和Paul Kingsbury。 命题库：一个带注释的语义角色语料库。 计算语言学，31（1）：71-106。

郝鹏，Sam Thomson和Noah A. Smith。 2017.用于语义依赖性解析的深度多任务学习。 在Proc。 ACL。

郝鹏，Sam Thomson，Swabha Swayamdipta和Noah A. Smith。 2018.从不相交的数据中学习联合语义解析器。 在Proc。 NAACL。

Jeffrey Pennington，Richard Socher和Christopher D. Manning。 2014. GloVe：单词表示的全局向量。 在Proc。 EMNLP。

Matthew E Peters，Mark Neumann，Mohit Iyyer，Matt Gardner，Christopher Clark，Kenton Lee和Luke Zettlemoyer。 2018.深层语境化词汇表示。 的arXiv：1802.05365。

Sameer Pradhan，Alessandro Moschitti，Nianwen Xue，Hwee Tou Ng，AndersBjo¨rkelund，Olga Uryupina，Yuchen Zhang和Zhi Zhong。 2013.使用OntoNotes进行强大的语言分析。 在Proc。 CoNLL。

Sameer Pradhan，Alessandro Moschitti，Nianwen Xue，Olga Uryupina和Yuchen Zhang。 2012. CoNLL-2012共享任务：在OntoNotes中建模多语言无限制共指。 在Proc。 EMNLP。

Vasin Punyakanok，Dan Roth和Wen-tau Yih。 2008.句法分析和推理在语义角色标注中的重要性。 计算语言学，34（2）：257-287。

Pranav Rajpurkar，张健，Konstantin Lopy-rev和Percy Liang。 2016.SQuAD：机器理解文本的100,000多个问题。 的arXiv：1606.05250。

Sunita Sarawagi，William W Cohen，et al。 2004.用于信息提取的半马尔可夫条件随机场。 在Proc。 NIPS，第17卷。

AndersSøgaard和Yoav Goldberg。 2016.深层多任务学习，在较低层监督低级别任务。 在Proc。 ACL。

Rupesh Kumar Srivastava，Klaus Greff和Ju¨rgenSchmidhuber。 2015.培训非常深入的网络。 在Proc。 NIPS。

马克斯蒂德曼 2000.信息结构和语法 - 音韵学界面。 语言探究，31（4）：649-689。

Swabha Swayamdipta，Miguel Ballesteros，Chris Dyer和Noah A. Smith。 2016.贪婪，使用Stack LSTM进行联合语法语义分析。 在Proc。 CoNLL。

Swabha Swayamdipta，Sam Thomson，Chris Dyer和Noah A. Smith。 2017.使用softmax-margin分段rnns和句法支架进行框架语义分析。 的arXiv：1706.09528。

OscarTa¨ckstrom，Kuzman Ganchev和Dipanjan Das。 2015.语义角色标签的高效推理和结构化学习。 ACL的交易，3：29-41。

谭志兴，王明轩，谢军，陈一东，史晓东。 2018.深度语义角色标注与自我关注。 在Proc。 AAAI

Kristina Toutanova，Aria Haghighi和Christopher D. Manning。 2008.语义角色标记的全球联合模型。 计算语言学，34（2）：161-191。

Ralph Weischedel，Martha Palmer，Mitchell Marcus，Eduard Hovy，Sameer Pradhan，Lance Ramshaw，Nianwen Xue，Ann Taylor，Jeff Kaufman，Michelle Franchini，et al。 2013. OntoNotes发布5.0 ldc2013t19。 语言数据联盟，宾夕法尼亚州费城。

Sam Wiseman，Alexander M Rush和Stuart M Shieber。 2016.学习共识解决方案的全局功能。 在Proc。 NAACL。

大卫H沃尔珀特。 1992.堆叠泛化。 神经网络，5（2）：241-259。

璧山杨和汤姆米切尔。 2017.用于帧语义解析的联合顺序和关系模型。 在Proc。 EMNLP。

Luke S Zettlemoyer和Michael Collins。 学习将句子映射到逻辑形式：用概率分类语法进行结构化分类。 在Proc。 UAI。

袁章和大卫韦斯。 2016.堆栈传播：改进了语法的表示学习。 在Proc。 ACL。

周杰和魏旭。 2015年。使用递归神经网络进行语义角色标记的端到端学习。 在Proc。 ACL。

A 补充材料

A.1 数据集

我们使用了FrameNet 1.5 re-lease8的全文部分[8]用于帧语义角色标记。 我们使用相同的测试集 达斯等人。 (2014)，并通过从列车集中选择8个文档来创建验证集。 该数据集包含3,139个具有16,621个目标注释的训练句子，387个具有2,282个目标的验证句子，以及具有4,427个目标的2,420个测试句子。 来自给定句子的每个目标被视为独立的训练实例。以下 Ta¨ckstro¨m等。 (2015)，我们只对每个具有多个注释的目标使用第一个注释。

我们使用OntoNotes中提供的标准拆分来进行CoNLL 2012共享任务。 该数据集包含115,812个火车句子，278,026个目标注释，15,680个验证句子，38,377个目标，12,217个测试句子，29,669个目标。

我们使用CoNLL 2012共享任务中的英语共指解析数据（Pradhan等人。，2012），包含2,802,343和348个文件，分别用于训练，验证和测试。

语法OntoNotes包含115,812个语法支架的训练实例。 FrameNet和OntoNotes训练数据之间没有重叠。

A.2 实验设置

我们使用GloVe嵌入（Pennington等。，2014）对于词汇表中的标记，随机词汇词汇被随机初始化。 对于frame-SRL，使用300维嵌入，并在训练期间保持固定。 对于Prop-Bank SRL，我们使用了在培训期间更新的100维嵌入。随后学习100维嵌入以指示目标位置 周和徐(2015)。 带公路连接的双向LSTM（Srivastava等人。，2015）使用6层之间，每层包含300维隐藏状态。 丢失0.1应用于LSTM。前馈网络的尺寸为150，深度为2，带有整流线性单元（奈尔和辛顿，2010）。 丢失0.2应用于前馈网络。

我们在FrameNet中将跨度的最大长度限制为D = 15，导致oracle在开发集上召回95%，在Propbank中召回13，导致oracle召回96%。 相同的最大跨度长度用于脚手架任务。 对于SRL支架，我们从OntoNotes中随机抽样实例以匹配SRL数据的大小，并在训练SRL批次和支架批次之间交替。 在FrameNet中，这相当于下采样OntoNotes。 对于Prop-Bank SRL，这相当于对OntoNotes的句法注释进行上采样，因为一个句子有一个句法树，但可以有多个目标注释，每个注释都是 - 姿态。

对于框架和PropBank SRL，混合比 δ 设定为{1.0（调整为0.1），0.5,1.0,1.5）。 我们用亚当（金马和巴，2014）用于优化，学习率为0.001，小批量为32。我们的动态程序公式也用于半CRF下的损失计算和推理。 为了防止爆炸渐变，渐变的2范数在渐变更新之前被剪切为1（格雷夫斯，2013）。 所有模型都训练最多20个时期，并根据dev F_1提前停止。

我们扩展了AllenNLP库，[9]是的建立在PyTorch之上。[10]每个实验都在一个TitanX GPU上运行。

对于共参照模型，我们使用相同的超参数和实验设置 李等人。 (2017)。 脚手架所需的唯一新的超参数是混合比 δ，我们根据验证集的性能将其设置为0.1。

[8]后来的版本，1.7也可用，但为了便于与其他已发布系统的比较我们报告了早期版本的结果。

[9]allennlp.org http://

[10]pytorch.org http://