## Experimental Setup

**Data.**我们在1.2.4版本的UCCA维基百科语料(UCCA Wikipedia corpus, 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(UCCA Twenty Thousand Leagues Under the Sea English-French parallel corpus, 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(<https://competitions.codalab.org/competitions/19160>)的官方网站以及github上找到。语料的详细数据详情见**表1**。对于依存句法任务，我们使用English Penn Treebank(PTB)作为语料。

**Evaluation.**为了评估两个UCCA DAG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$yield\left(u\_{p}\right)=yield(u\_{g})$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。我们将使用官方提供的脚本(<https://competitions.codalab.org/competitions/19160>)来评估primary edges以及remote edges的F值，标点符号仅在评估时被排除在外。

**Closed Settings.**我们仅仅使用UCCA语料以及预训练词向量([http://fasttext.cc](http://fasttext.cc/))在UCCA单任务学习中，没有使用任何语料中提供的自动特征。我们探索了以下几种模型的变化：(1)使用BiLSTM或者self-attention作为编码器。(2)使用top-down loss配合top-down解码以及local loss配合CKY解码。这就导致了总共四种不同的模型结果进行比较。

**Syntax-enhanced Settings.**我们直接采用了目前效果最好的biaffine parser作为句法的基础模型，其中为了保持和UCCA模型的输入一致自动词性特征被改成了相同的charlstm。经过训练后，parser在PTB上F值达到了94.39%，然后利用它生成UCCA语料中的依存句法树作为辅助的资源。

**Use of BERT.**为了进一步探究句法对于UCCA的作用，我们从BERT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后拼接在词向量之后输入到上层的编码器。

**Training.**对于所有的实验，都采用了Adam优化器，学习速率为0.001，并且裁剪正则大于5的梯度。我们设置迭代的最大次数为100，当开发集上的F值经过15次迭代后仍然没有提升，则训练提前结束。在多任务学习框架时，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乘上他们训练集之前的相应倍数，确保两个训练集在相同的batch数中使用完全。

## Results