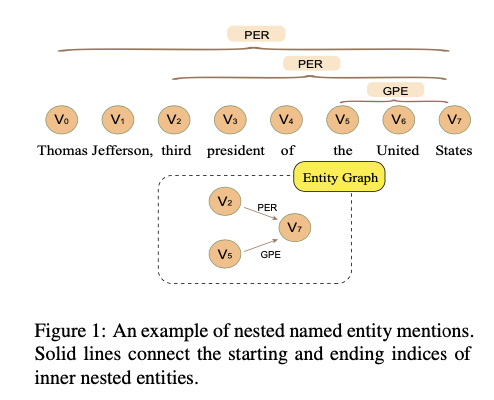
论文阅读笔记一

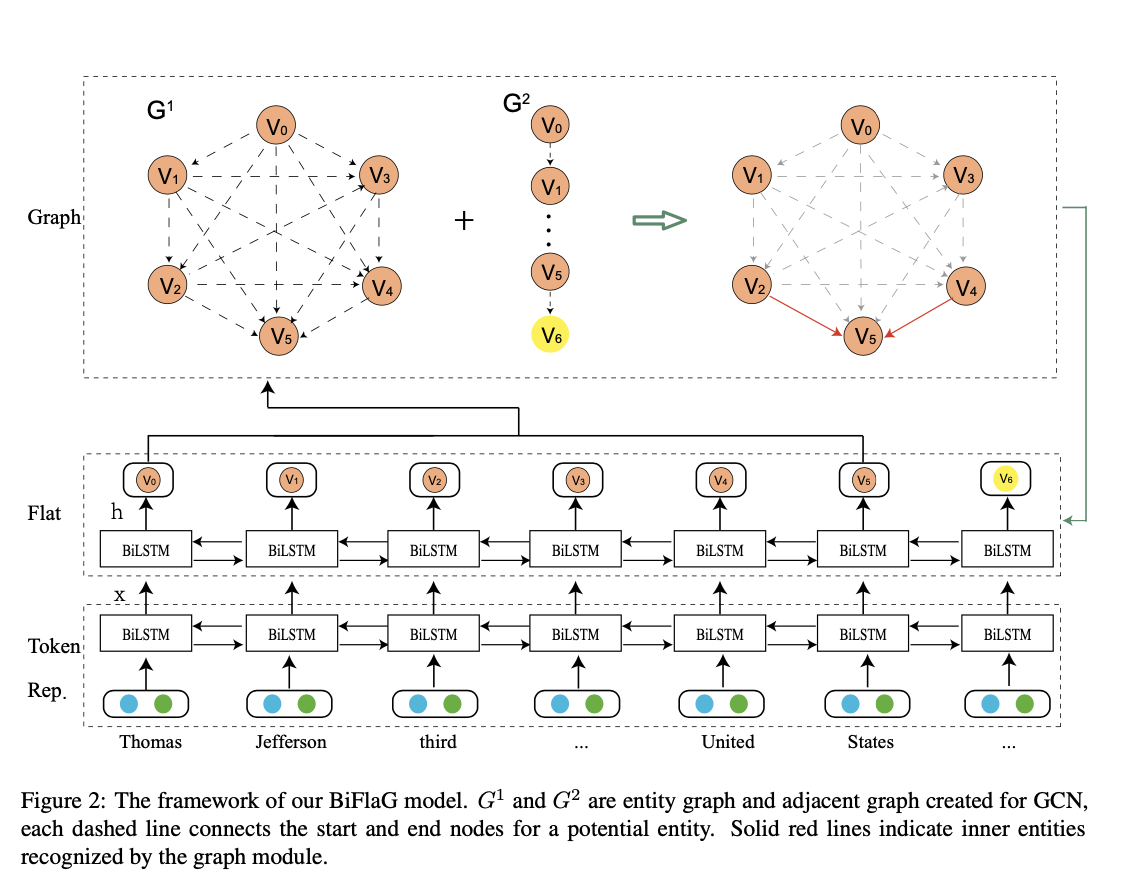
Bipartite Flat-Graph Network for Nested Named Entity Recognition

1. 贡献
2. 提出了一种新的用语嵌套实体识别的二分平面图网络BiFlaG，它包含用于识别最外层非嵌套实体的模块和用于识别内层实体的图模块；
3. BiFlaG充分利用最外层实体的序列信息，同时也考虑了外层和内层实体的交互信息；
4. 方法

存在嵌套实体的情况如下：其中最外层的PER为flat entity，第二层的PER和第三层的GPE为嵌套实体。



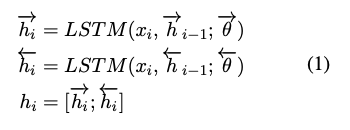
模型整体结构如下：模型用BiLSTM+CRF来处理外层的平面实体，用一个图模型来处理嵌套实体。



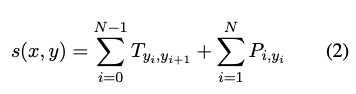
1. 用BiLSTM处理序列中的Flat entity

给定包含N个token的序列{t1,t2,…,tN}，首先将每个token字符级向量Ci和单词级向量wi拼接起来得到ti=[wi; ci]；

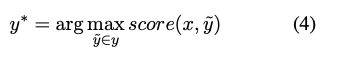
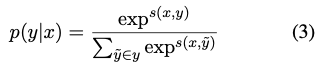
然后用BiLSTM获取每个token的序列信息：xi=BiLSTM(ti)，将xi作为输入送到之后的模块中：

其中θ是待训练的参数，h是前向和后向隐藏层的向量。

将BiLSTM学到的序列信息h送到CRF中学习标签之间的依赖：

其中T表示转移概率，P表示发射概率

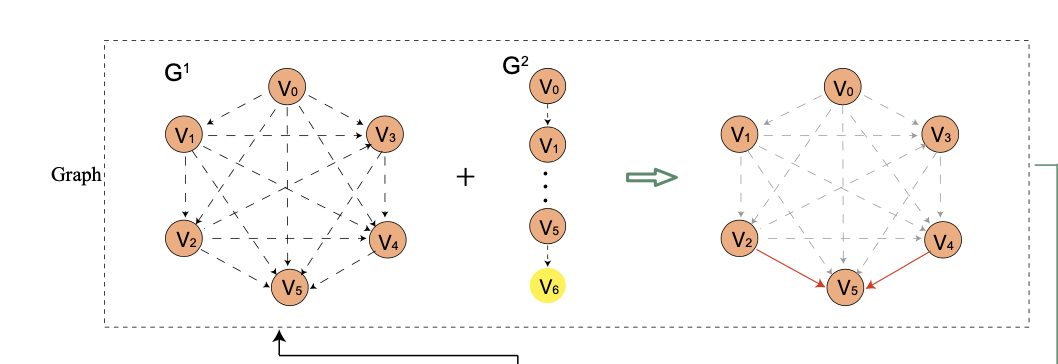
CRF模型通过最大化给定输入x条件下输出y的概率来进行预测：



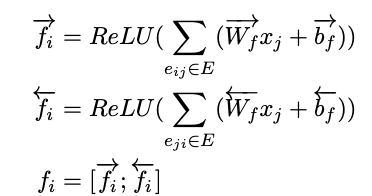
1. 用图模块来处理Nested entity

首先根据flat entity模块提取到的序列信息构建图，然后用GCN来提取内层实体。

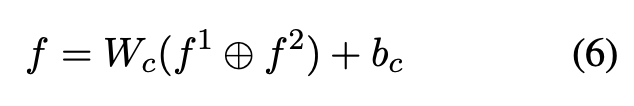
1. 实体图G1：flat 实体模块识别到的所有节点和边都将被加入到图中，eij={vi, vj}，其中start≤i≤j≤end。该图保证最外层的flat实体信息也被用到；



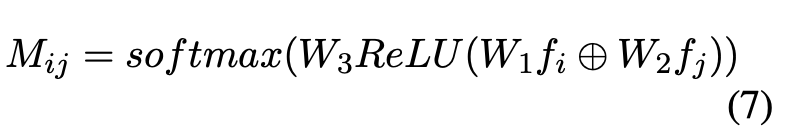
1. 邻接图G2：对每对相邻的单词，都加入一条从左到右的直接边，以此来利用局部上下文信息。
2. Bi-GCN：考虑每个节点的输入和输出信息，用Bi-GCN提取图特征，给定图G=(V, E)和单词的向量表示X={x1, x2, …, xN}，通过非线性激活函数ReLU学习图特征f：

其中，wf和bf是待训练的参数

G1和G2的图特征将被整合到一起：

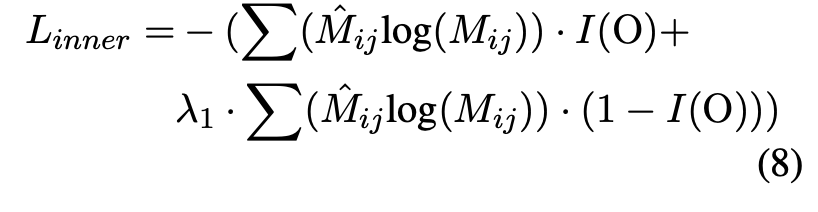


获取到 图的特征表示F={f1, f2, …, fN}之后，再对内层实体进行打分：



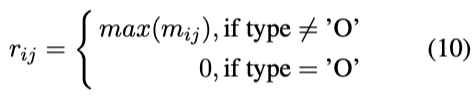
Mij表示从第i个token到第j个token为某一类型实体的概率。

用交叉熵函数来计算损失：

，λ1越大，实体的影响越大，O类型实体的概率越小。其中I(O)表示实体类型为O或者不为1.

1. BiGlaG模型的训练

公式(7)表示句子中每个单词对为某类实体的概率，用最大池化操作的结果rij表示从token i到j表示该实体为实体或是非实体:

如果从i到j没有依赖，每个词都为O，则rij为0，否侧为max(mij)

由此可以得到一个包含内层实体边界信息的图G3=(V, E)，其中rij∈E。

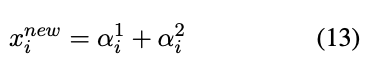
用于更新flat模块的表示由两部分组成：一是每个token之前的表示：



第二部分汇总了新图G3的内部实体依赖性：



最后将两个部分的内容汇总到一起：



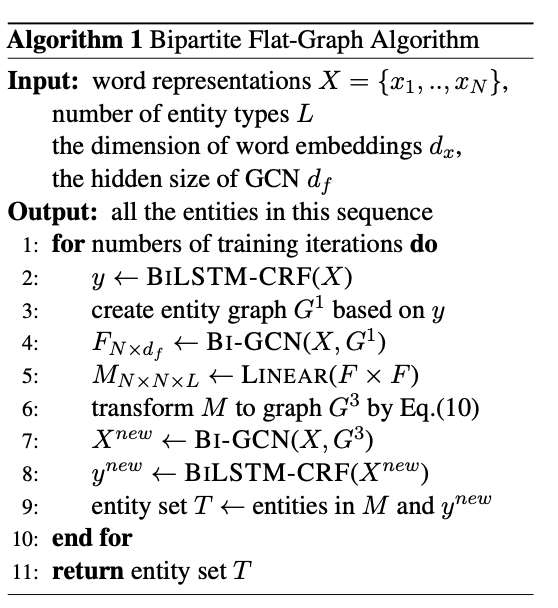
新的x将被送到flat模块去更新外层的参数，提取更好的外层实体信息。

语料采用BIOES标注法，用CRF来计算损失：



语料中的实体被分为外层实体和内层实体，分别用两个模块来提取实体，每个模块的实体都共享相同的神经网络结构。

模型的整体训练过程如下所示：



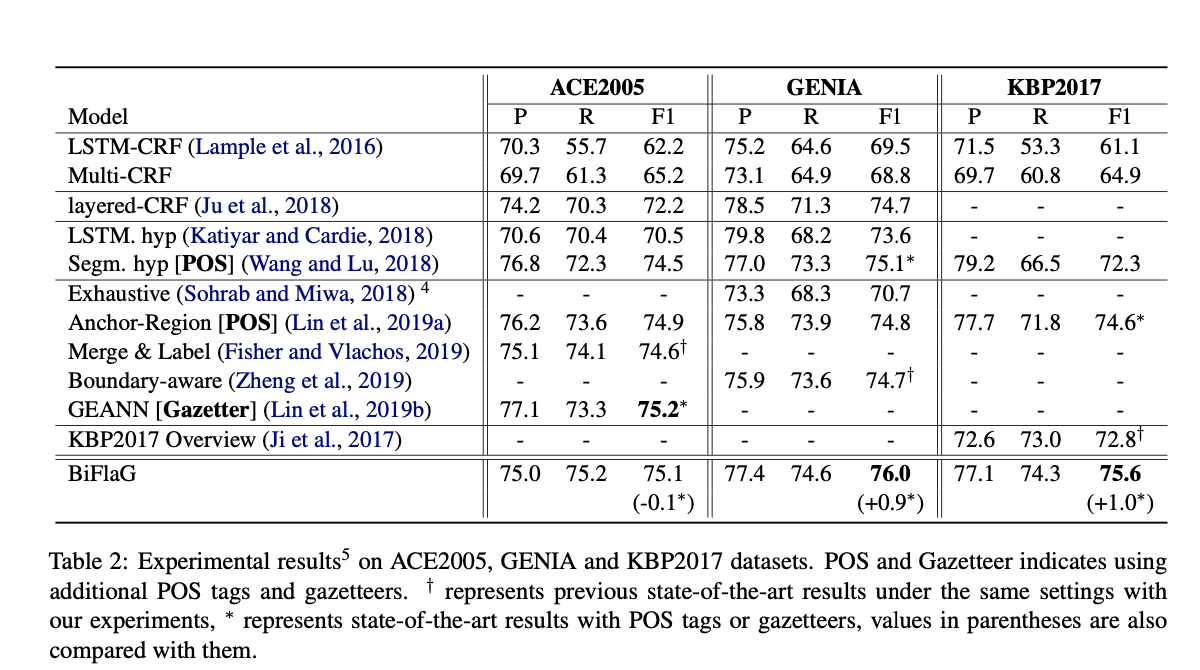
BIFLaG模型预测内层实体的时候，总的损失函数为：

 λ2为flat模块与图模块之间的损失的权重，训练目标为最小化总的损失函数。

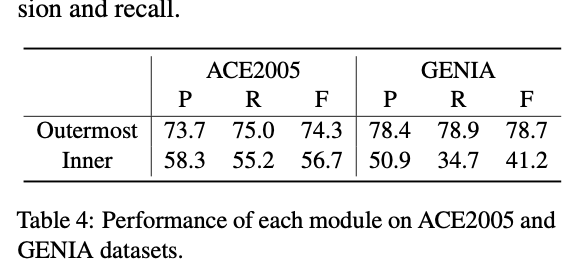
1. 实验

实验数据：ACE2005、GENIA、KBP2017，分布包含22%、10%、19%的嵌套实体。

衡量指标：precision、recall、f-score



1. 实验结果分析



作者也评估了这两个模块作用于ACE2005和GENIA的效果，flat模块在两个数据集上的外层实体识别上都有很好的效果，但是在GENIA的内层实体上的recall值很低。

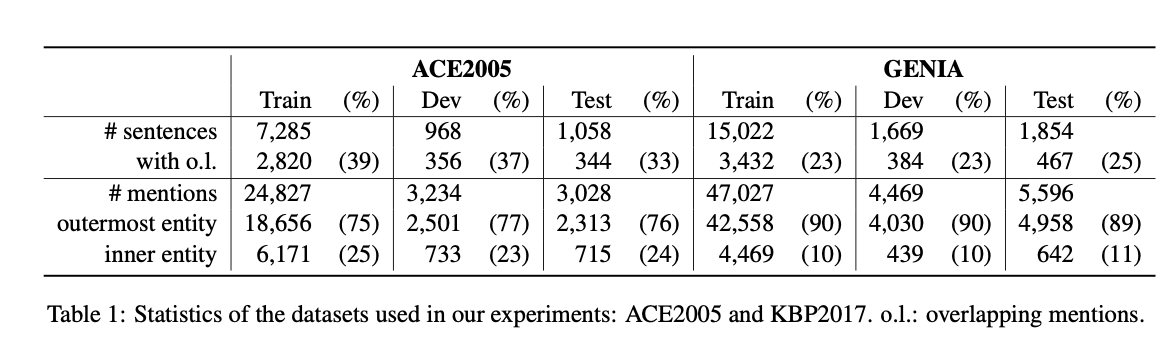


Table1数据表明，GENIA中的内层实体占比仅11%，而ACE2005中有24%。

可以通过改变公式(15)中的λ1的值来改变内层实体稀疏性的影响，实验显示，λ值在1.5时能够平衡precision和recall之间的影响。

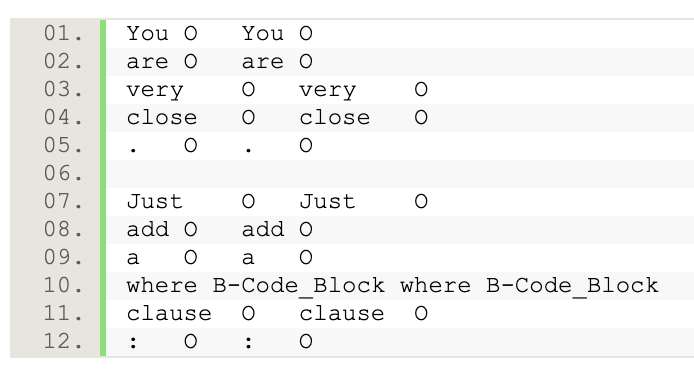
论文阅读笔记二

Code and Named Entity Recognition in StackOverflow

1. 贡献
2. 发布了一个由20种带标注的实体的语料库
3. 一个可以识别20种软件开发相关的实体的域内训练神经NER标注器
4. 一个代码token识别器，利用StackOverflow 上代码块去获取与代码相关的实体的字符模式，并不断改进NER 标注器。



语料库介绍：共1237个问题，一个问题+四个答案（1accepted+1得票最高+2随机）。同时也标注了 GitHub上从readme文件和issue report中的6501个句子，将这些数据作为额外的评估数据。

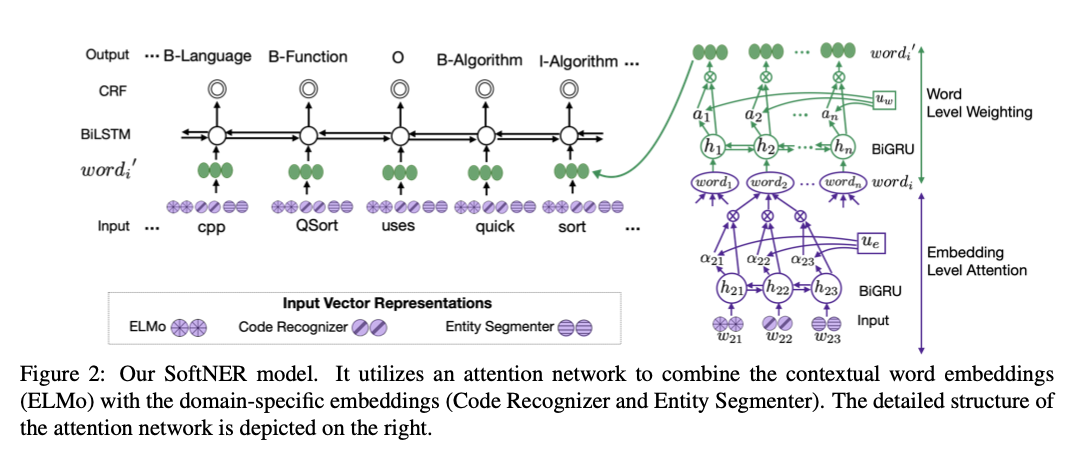


每行的字符和标签用”\t”隔开，NE是实体标记，<markdown>是发帖用户的代码块标记。

语料总共包括20种实体，其中8种与代码相关，12种与自然语言相关。代码相关的实体为：CLASS、VARIABLE、IN LINE CODE、FUNCTION、LIBRARY、VALUE、DATA TYPE、HTML XML TAG；自然语言实体包括：AP- PLICATION、UI ELEMENT、LANGUAGE、DATA STRUCTURE、ALGORITHM、FILE TYPE、FILE NAME、VERSION、DEVICE、OS、WEBSITE和USER NAME。

1. 方法

SoftNER：提取与软件相关的命名实体。该模型包括三个部分，一是embedding提取层，该层为输入句子中的每个单词创建上下文化的ELMo嵌入和两个新的特定域的嵌入；二是多层次注意力层，使用embedding级别和单词级别的注意力网络将三个单词嵌入组合在一起；三是BILSTM-CRF层，该层使用来自上一层的加权单词表示来预测每个单词的实体类型。



1. Input Embedding

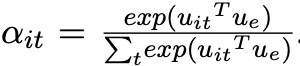
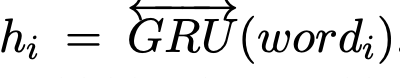
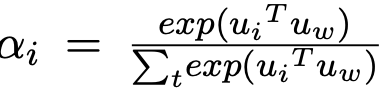
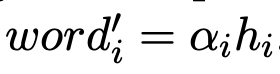
对于输入的句子，每个句子可能都包括编程语言和自然语言，作者提取数据ELMo向量表示和两种新的特定领域的向量表示。一种是用代码识别器（code recognizer）生成的向量，表示一个单词是否为一个代码相关的实体（CLASS、VARIABLE等；另一种是实体分割器生成的向量，它预测一个词是否为给定句子中的任意一个实体的一部分。

1. 代码识别Code Recognizer：这是一个二分类模型，最终产生Code Recognizer向量。它利用语义模型特征和字符模板去预测一个单词是否为一个代码实体，它用到的特征包括在Gigaword语料库上训练过的两种语言模型（LM）的unigram单词和6-gram字符的概率，以及StackOverflow 10年档案库中的所有代码片段。同时也用这些代码块预训练了Fastext词向量。

首先用高斯分割将每个ngram概率转换成k维向量，然后将向量化的特征送到线性层，将它的输出与 FastText的字符级向量连接起来，然后将他们送到有sigmoid激活函数的隐藏层。如果输出的概率大于 0.5，则输入的token是一个代码实体。

1. 实体分割Entity Segmenter：识别每一个token，判断它是否为实体，是实体则标记为E-ENTITY，否则标记为O。这个分割模块用BiLSTM做encoder，用CRF做decoder，输入包括ELMo嵌入、词频和代码的markdown标签。
2. Multi-level attention layer

使用多级注意力网络去计算每个单词对任务的重要性。

1. Embedding-level Attention：注意力网络的第一层，将词嵌入组合在一起，对于输入的每一个词wi，使用wi1(ELMo)、wi2(code recognizer)、wi3(Entity Sementer)三种词嵌入；注意力网络的第二层，计算每个单词对任务的重要性，为了计算含有注意力信息的嵌入αit，将组合之后的词嵌入送到注意力网络中，通过双向GRU生成相应的隐藏层向量表示，然后将隐藏层向量hit送到一个非线性层得到输出uit =tanh(Wehit + be )，引入上下文向量ue，将ue与隐藏层输出表示组合，提取嵌入的权重，。最后，通过不同嵌入的权重信息之和来生成词向量：
2. 含有权重的单词表示：使用单词级别的权重参数αi来表示每个单词对任务的重要性。参数αi的计算与Embedding-level的注意力相似，此时模型的输入为词向量wordi。第一步计算隐藏层向量，第二步计算隐藏层输出，第三步提取权重，其中uw是待训练的参数，最后计算含有权重表示的词向量
3. BiLSTM-CRF

将上一步得到的送到BiLSTM-CRF中进行预测。

1. 实验参数和结果
2. 参数

在PyTorch框架下实现，输入为500维的分割向量、300维的代码识别向量和1024维的上下文词向量表示，用Glove、ELMo、BERT在来自StackOverflow的十五亿句子（除了带标注的1237个句子）上训练领域内的词向量表示。

分别训练SoftNER模型和两个辅助模型（代码识别与实体分割）。

1. 结果

