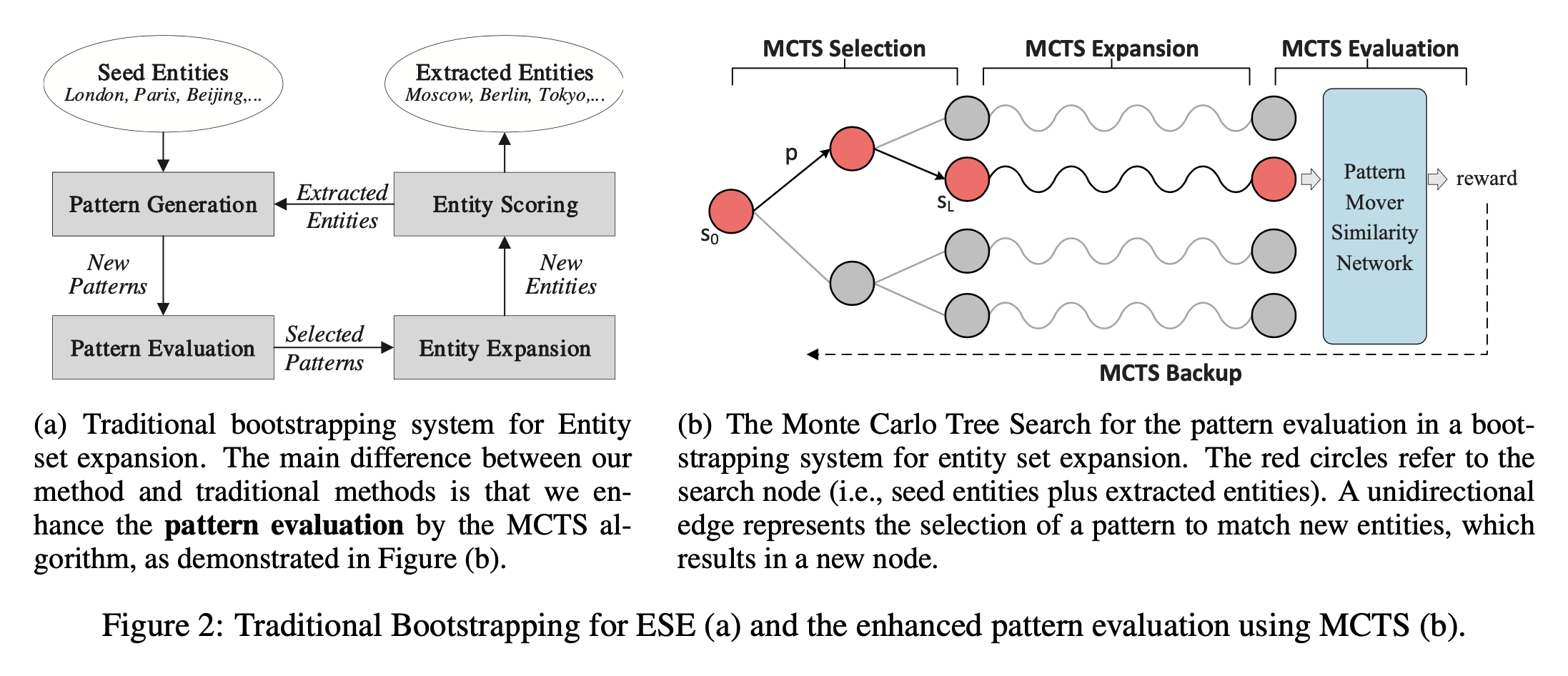
**论文阅读笔记一**

Learning to Bootstrap for Entity Set Expansion

源码：<https://github.com/lingyongyan/mcts-bootstrapping>

1. 问题与贡献
2. 传统的Bootstraping方法通常有延迟反馈（模型评估依赖于其直接提取的信息和之后迭代的信息的质量）导致的语义漂移和稀疏监督（只有少数种子实体）的问题
3. 本文提出一种新的bootstraping方法，这种方法将MCTS算法与深度相似性网络组合起来，可以有估计估反馈延迟从而行模型评估，并且对给定稀疏监督信号情况下对实体进行自适应评分。
4. 贡献
5. 通过结合MCTS算法的增强型Bootstraping方法评估bootstraping的延迟反馈(迭代反馈)；
6. 用一种新的深度相似性网络，用于在Bootstraping中评估实体集扩展的实体；
7. 结合深度相似性网络和MCTS算法的自适应实体评分
8. 方法

在传统bootstraping方法的pattern evaluation和entity score阶段进行优化，分别用MTCS算法和PMSN。



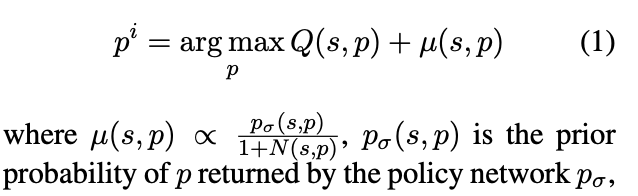
|  |  |
| --- | --- |
| **Figure 2: (a)传统Bootstraping算法流程** | |
| 1. Pattern generation | 给定种子实体和提取到的实体，Bootstraping方法从语料库中生成pattern，本文用词汇-句法作为模板。如Beijing is the capital of。 |
| 1. Pattern evaluation | 评估种子实体产生的pattern，通常使用RlogF函数来打分，主要评估每个pattern的即时反馈。 |
| 1. Entity expansion | 选择得分高的若干个pattern去匹配新的候选实体。 |
| 1. Entity scoring | 用bootstraping方法对候选实体进行评分，得分最高的实体将被加入实体集。 |

存在的问题：第二步时未考虑迭代过程中产生的延迟反馈；第四步评分时，由于稀疏监督造成的不可靠评分也反过来影响第二步的延迟反馈评估。

|  |
| --- |
| **本文改进的Bootstraping算法** |
| 1. Pattern generation |
| 1. Pattern evaluation + Monte Carlo Tree Search(MCTS) |
| 1. Entity expansion |
| 1. Entity scoring + Pattern Mover Similarity Network（PMSN） |

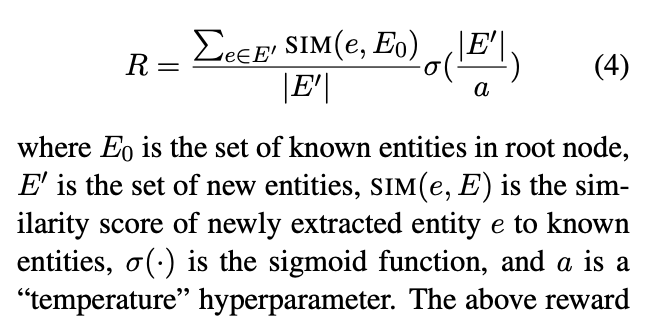
**第二步中，用MCTS缩小搜索空间，解决延时反馈导致的语义漂移问题。**

**MCTS算法流程如下（如Figure2: (b)所示）：**

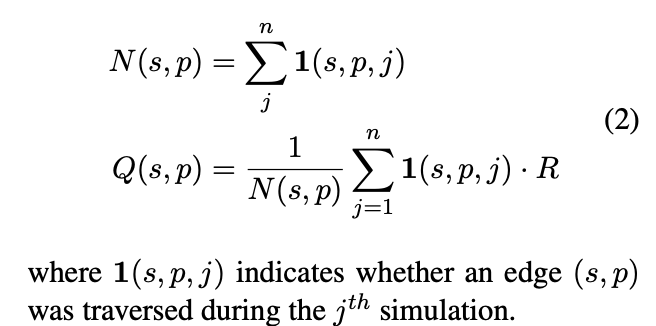
1. Selection: 节点开始，每次都遍历MCTS直到到达叶子节点或是固定深度，每步i都通过以下方式选择：

Q(s, p)是节点s选择p的累计分数，N(s, p)是遍历(s, p)的次数。这一步选择高分或是高先验概率的pattern，可以缩小搜索空间。

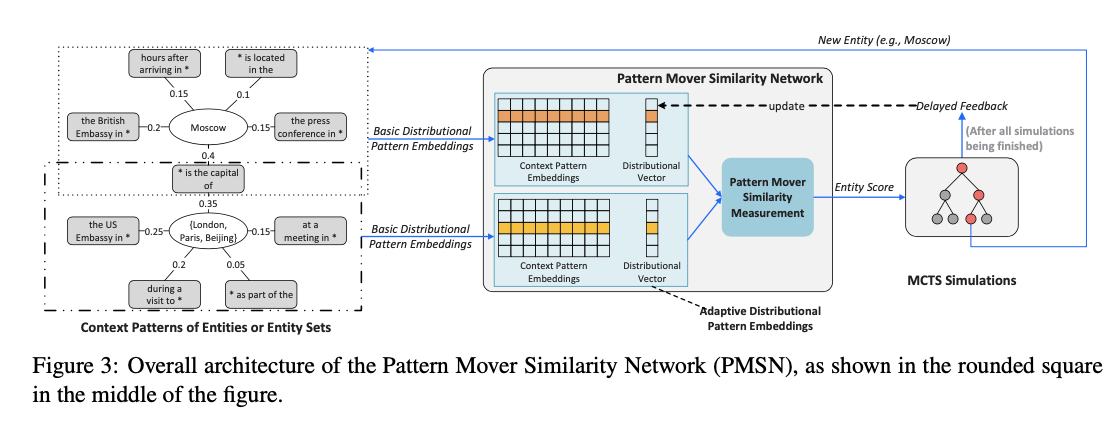
1. Expansion：当遍历到叶子节点的时候，通过选择新的pattern去匹配实体。因为新生成的pattern缺乏累计分数，所以用先验概率。
2. Evaluation：当第二步结束或达到固定深度之后，用RlogF快速选择pattern以扩展节点，然后在返回节点的奖励R以评估新提取的实体的质量。



1. Backup：将前面返回的节点的奖励R用于更新(s,p)的累计奖励和访问次数：

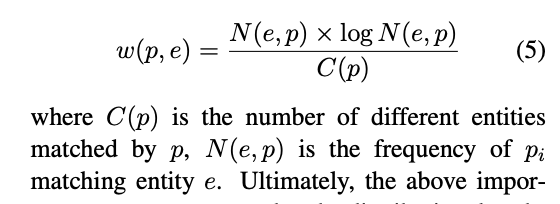
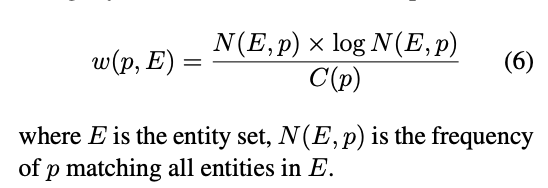


MCTS过程得到的累计奖励Q(s,p)将被用做pattern p的延迟反馈，选择累计奖励高的pattern更容易得到正确的实体。

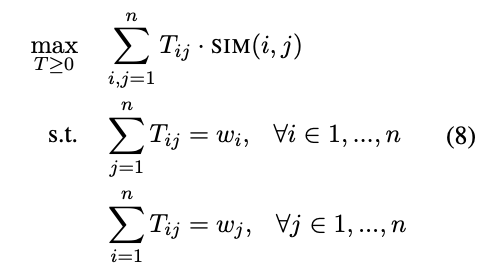


**第四步，用PMSN统一对实体、pattern、实体集进行嵌入，评估他们的向量相似性。**该网络包含两个部分，一是自适应分配pattern向量；二是向量相似性评估。

**第一个部分：自适应分配pattern向量的生成**

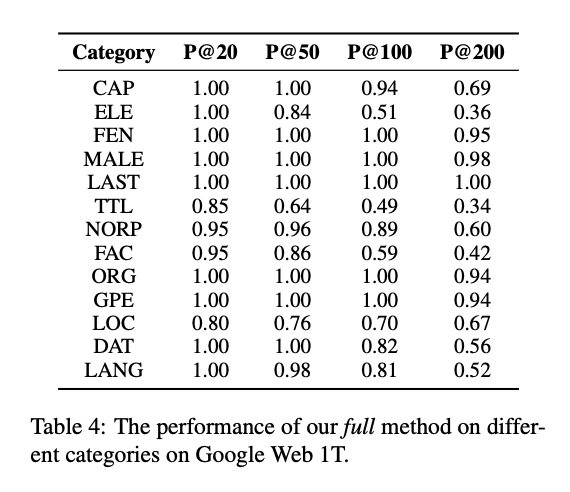
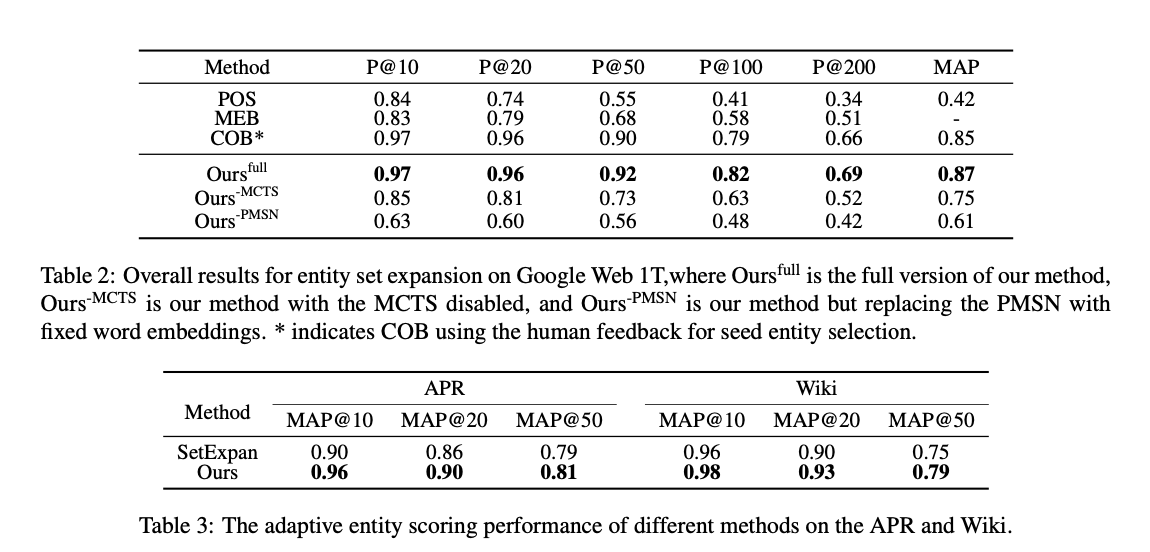
首先用预训练的GloVe将pattern表示为上下文pattern的单一嵌入；然后根据语言学的分布假设，用分布pattern向量（DPE）存储不同pattern的重要性得分和上下文pattern嵌入，以此来表示实体的语义(Figure3 中间的圆角方框)，对e的每个p打分，在实体集E上对p打分；最后用<X, w>来表示DPE，其中X表示上下文pattern嵌入矩阵，w表示概率；最后在DPE 的基础上，将MCTS与PMSN相结合（Figure3最右MCTS反馈部分），形成自适应分配pattern向量（ADPE）。

**第二个部分：向量相似度的计算**

****，SIM(I,j)表示两个实体的第i和第j个pattern嵌入的余弦距离。

1. 实验结果

实验数据：Google Web 1T 、APR和Wiki

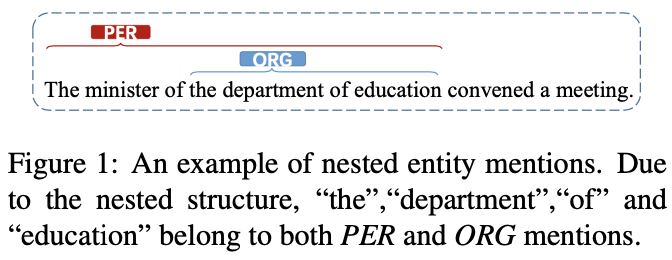


**论文阅读笔记二**

源码：<https://github.com/sanmusunrise/ARNs>

Sequence-to-Nuggets: Nested Entity Mention Detection via Anchor-Region Networks

1. 问题、相关工作与贡献
2. 问题：基于序列标注的命名实体识别限制了一个词只能属于一类实体，当遇到嵌套实体时就会面临一系列问题。

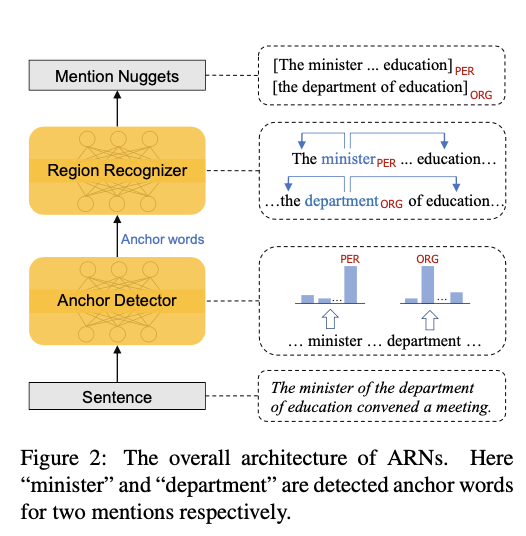


1. 相关工作

Region-based approach：通过划分子序列来识别嵌套实体，需要直接对子序列进行分类，计算复杂度较高；

Schema-based approaches:通过设计更加强大的标注架构来识别嵌套实体，但架构的识别能力越强，结构也月复杂，时间复杂度也更高。

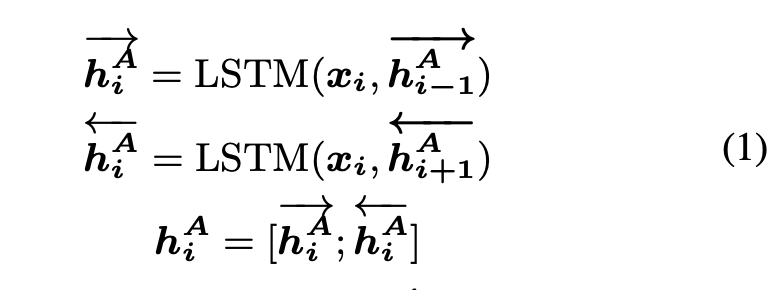
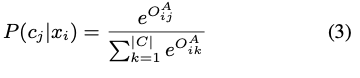
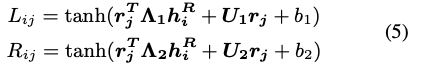
1. 主要贡献
   1. Anchor-Region Networks(ARN)，一种为识别嵌套实体而设计的sequence-to-Nuggets结构，将嵌套命名实体的识别看作两步：anchor word识别和anchor word对于实体的边界的识别，通过建模和利用实体的head-driven短语结构来解决嵌套实体探测问题。
   2. Bag Loss，利用单词和实体类型之间的关联，以端到端的方式学习ARN。无须anchor Word。
   3. 提出一种基于神经网络的方法，用于挖掘Head-Driven短语结构。
2. 方法

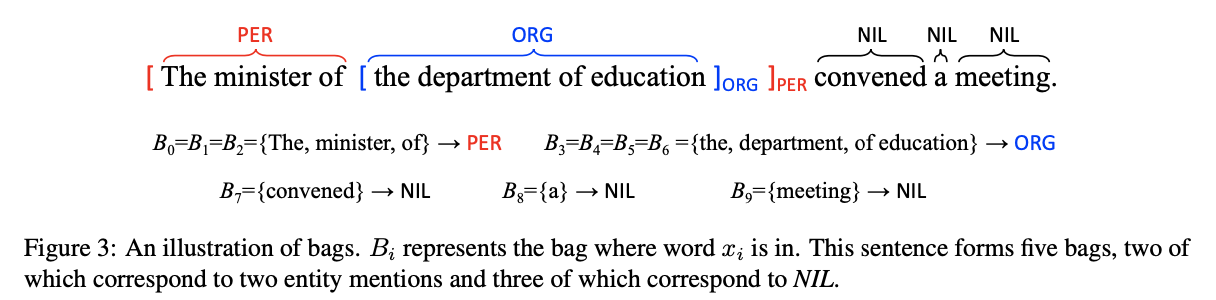


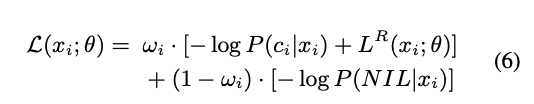
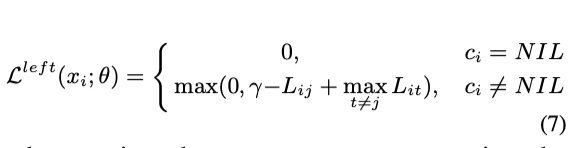
第一步：Anchor Detector，一个单词级别的分类器，找出anchor Word并且判断其对应的实体类别；比如“the minister of the..”，minister为PER类别实体的anchor word。

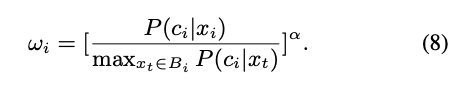
第二步：Region Recognizer，一种pointer-based的结构，确定mention nuggets，确定anchor word为中心的实体的边界；如“minister”的mention nugget是“the department of education”。

**算法流程（1-3 anchor detector，4-6 region recognizer，7-模型学习&bag loss）**

1. 将给定句子映射为词的向量表示，本文中用了Word embedding、part-of-speech embedding和character-based representation of word，包含了词的语义信息和字符特征等信息。
2. 然后用BiLSTM获取每个词的上下文特征表示：
3. 将送入多层感知机分类器中进行多分类，得到词为某类实体的anchor Word的分数：
4. 用softmax对分数进行归一化：。不同的实体不会共用相同的anchor Word，所以anchor detector为不同的实体识别不同的anchor word来解决嵌套实体识别问题。
5. 用CNN提取句子局部特征，，其中k为窗口大小，w和b为卷积核和偏置项。
6. 计算anchor word 在anchor word 中的左、右边界的分数：
7. 取得分最高的左边界、右边界，得到Nugget,。
8. 模型学习——bag loss

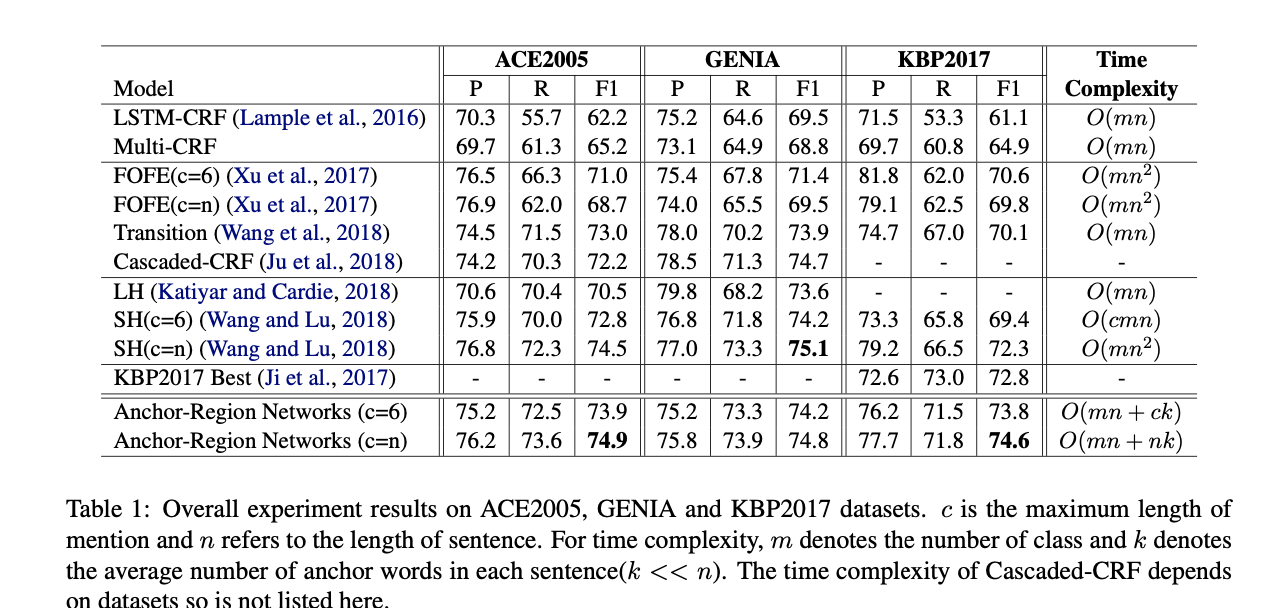


，其中为anchor detector loss，为实体边界损失region recognizer loss，其中左边界的损失定义为，其中γ为超参数。右边界的损失函数类似。

而权重用于衡量anchor word和bag类型之间的相关性，关系越紧密则越大：其中α表示一个词被视为anchor word的可能性。

1. 实验结果

ACE2005、GENIA、KBP2017在ACE 2005和GENIA数据集中各有20%、10%左右的嵌套实体。



1. 实验结果分析

LSTM-CRF和multi-CRF对比说明嵌套实体对于NER的识别有很大的影响；

Anchor-Region Networks在三个数据集上都能取得很好的效果;

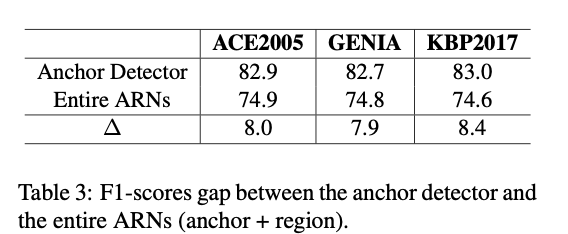
Head-driven phrase structure of entity能够降低计算复杂度。

1. Bag loss 的作用

因anchor word的存在，所以bag loss是有效的；

当α(式8)大于特定阈值的时候，bag loss对α不再敏感；

1. 错误分析



Anchor detector的F1比完整的ARN高8%左右，错误主要是由于region detector导致的。比如：

正确的边界应该在old。可能因为缺乏语法知识导致边界划分出错。

可以考虑在模型中加入句法、语法知识来解决这个问题。