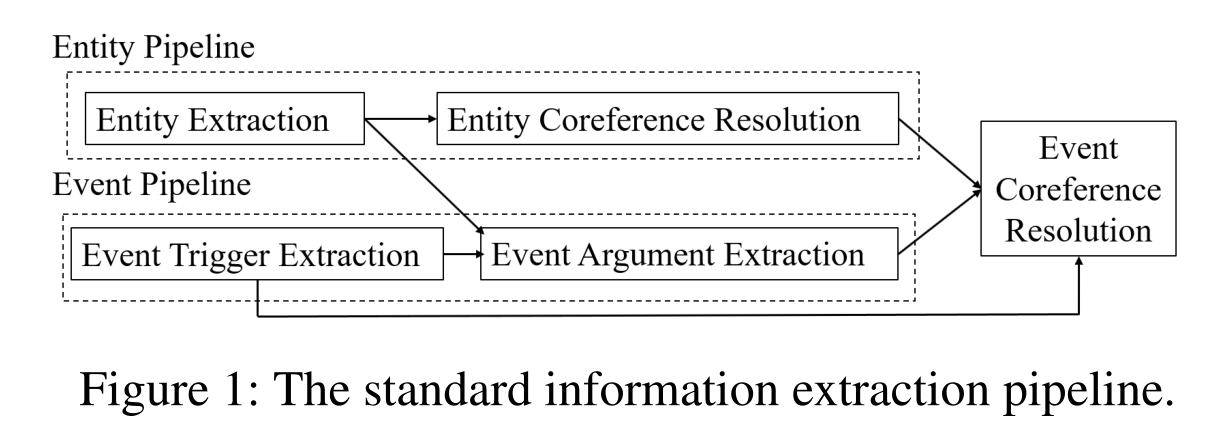
时间：20201/8/2

*(2021 AAAI)Span-Based Event Coreference Resolution*

基于span的模型最近已成功应用于基于实体的IE任务，包括实体提取和实体共指解析。与许多其他神经模型不同，基于span的模型专注于学习特定任务的span，而不是单词表示。例如，当应用于实体共指时，基于span的模型寻求学习对应于实体提及的文本span的表示，使得两个共指实体提及具有相似的span表示。传统的实体共指研究集中于设计复杂的共指模型，而基于span的模型集中于学习可以与相对简单的共指模型结合使用的span表示。



事件共指消解比实体共指消解更具挑战性。它涉及(1)从给定文档中提取实体提及(实体提取组件)，并确定其中哪些是共指的(实体共指消解组件)；(2)通过识别它们的触发词/短语并确定哪个实体提及是它们的参数/参与者来提取事件提及(事件提取组件)；以及(3)使用来自实体管道和事件管道的信息来确定哪些事件提及是相关的。

受最近基于span的模型成功应用于基于实体的信息提取任务的启发，我们研究了基于span的事件共指消解模型。

三个研究问题

1、首先，基于span的模型对于事件共指解析的效果如何？

2、在基于span的事件共指模型中，利用事件共指和触发器检测之间的依赖性的最佳方式是什么？

3、实体共指信息有利于事件共指解决吗？

贡献

* 研究了基于span的模型在事件共指解析中的应用，以及跨任务一致性约束和实体共指信息在基于span的事件共指解析中的利用。
* 在KBP 2017事件共指数据集上的结果证明了基于span的事件共指模型的有效性，尤其是当增加了一致性约束和实体共指信息时。

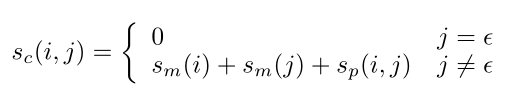
对于第一个研究问题：

首先设计了一个基于span的事件共指模型，作者将把它用作基线，并在后续部分中增加跨任务约束和实体共指信息。该模型将文档D作为输入，文档D表示为单词标记序列，从文档D中提取所有可能的句子内跨度，最大长度为L。每个这样的跨度对应于一个候选触发器，并且该模型同时学习触发器检测和事件共指解析。

**触发检测任务**旨在为每个跨度I分配一个子类型yi，每个YI在子类型清单中取一个值或NONE，表示I不是触发器。该模型预测I的子类型为y∫I = arg maxytst(I，yt)，其中STI是一个评分函数，表明I具有yi子类型的可能性。

IMG_256

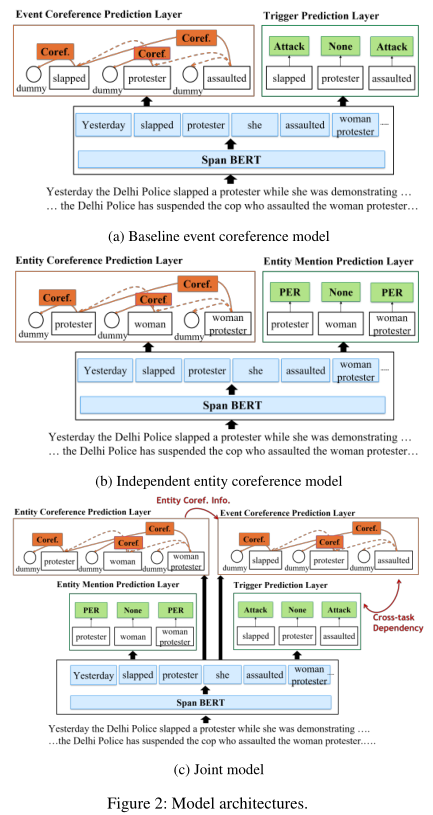
**事件共指消解任务**旨在为span i分配一个先行词yc，其中YC∞{ 1，.。。，I-1，？}.换句话说，每个yci的值是其先行词的id，它可以是前面的跨度之一，也可以是一个虚拟先行词？。我们定义了以下评分函数:



其中sm(i)是表明I成为触发器的可能性的分数，sp(i，j)是在I和前一个跨度j上计算的成对共指分数。该模型预测I的前件为Y∫c = arg maxj∈Y(i)sc(I，j)，其中Y(I)，I的候选前件集，包含相关文档中I之前的所有跨度。

IMG_256

**模型结构**

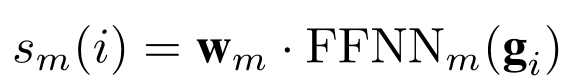


**Span Representation Layer.**

首先，将输入文档分成不重叠的区域，每个区域中的单词序列用作输入训练序列；然后，将序列传递到SpanBERT-large 中的预训练转换器编码器中，对标记及其上下文进行编码；最后，span i的表示设置为gi：

IMG_256

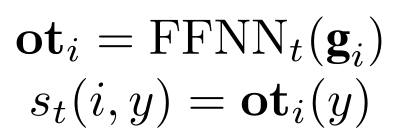
hstart(I)和hend(i)是span的开始和结束标记的隐藏向量，hhead(i)是基于注意力的头部向量，fiis是span宽度特征嵌入。



其中FFNN是前馈神经网络。然后，我们只保留跨度的前N%用于进一步处理。

**Trigger Prediction Layer**

将span i的表示gi传递给一个FFNN，FFNN输出一个T维向量 oti，其中T是可能的事件子类型的数量(包括NONE)。oti(y)，oti的yth元素，是一个分数，表示I属于事件子类型y的可能性。



**Coreference Prediction Layer**

首先计算跨度I和j之间的成对得分，

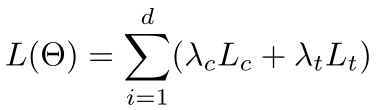
IMG_256

其中♀表示元素式乘法，gi♀gj编码跨度I和跨度j之间的相似性，uij是编码两个跨度之间距离的特征嵌入。

我们可以使用等式2和5计算等式1中定义的共指得分。

**Training**

损失函数L(θ)由两个任务的损失组成，定义如下:



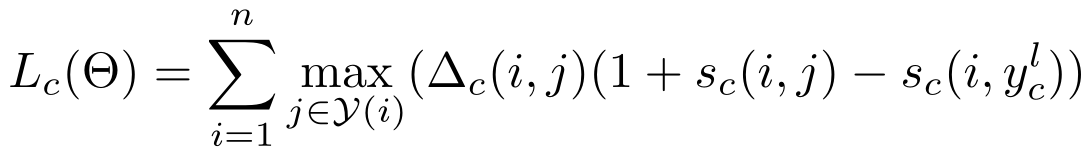
**Task Losses**

对每项任务采用最大利润损失。

作者采用怀斯曼等人(2015)定义的**共指损失函数**来解决实体共指。具体来说，让GOLDc(i)表示跨度I之前与I相关的跨度集，ylc是argmaxy∈GOLDc(i)sc(i，y)。换句话说，ylc是I的最高得分(潜在)前因。

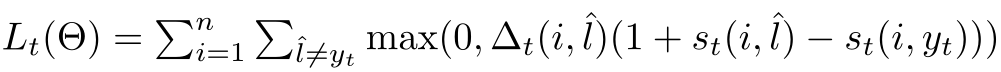
IMG_256

共指损失函数定义为:



其中Δc(I，j)是一个特定错误的成本函数，它返回与特定类型的错误相关的成本。通常，如果预测的先行项j的得分高于正确的潜在先行项yl c，则损失函数会惩罚跨度I。

我们类似定义了**触发检测的损失:**



通常，损失函数会惩罚每个错误子类型l的分数高于根据st的正确子类型yt的每个跨度。

**Exploiting Cross-Task Dependency**

虽然基线采用的多任务学习设置允许触发器检测和事件共指通过共享表示层相互受益，但它未能利用它们之间存在的依赖性。下面我们研究利用这种依赖性的四种方法。

**Gold feature (GF)** 作者使用一个额外的特征训练基线共指模型，该特征编码所考虑的两个事件提及是否具有相同的子类型。**在训练期间**，我们使用黄金子类型计算这个特征。**在测试期间**，作者使用触发检测器预测的子类型来计算这个特征。

**Predicted feature (PF)** 本质上与第一种方法相同，除了使用触发检测器预测的事件子类型在**训练和测试期间**计算特征。

**Hard constraint (HC)** 在前两种方法中，触发器检测和事件共指之间的依赖关系被隐式编码为一个特征，因此事件共指模型有可能将具有不同子类型的两个事件提及假设为共指。我们的第三种方法**将这种依赖性显式地编码为硬约束**。具体来说，给定要解决的事件提及，在训练期间过滤其黄金子类型不同于事件提及的所有候选前因。**在测试期间，**我们根据预测的子类型进行筛选。

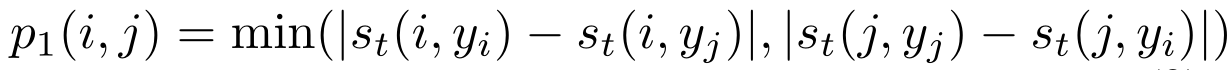
**Soft constraint (SC)** 探索了一种新的方法来编码这种与一致性约束相关的依赖关系。具体来说，我们在基线中加入了对触发器检测和事件共指输出的以下两个一致性约束。

P1：如果两个跨度没有相同的事件子类型，它们不能是共指的。

P2：如果一个跨度的事件子类型为NONE，则其先行词必须是虚拟先行词。

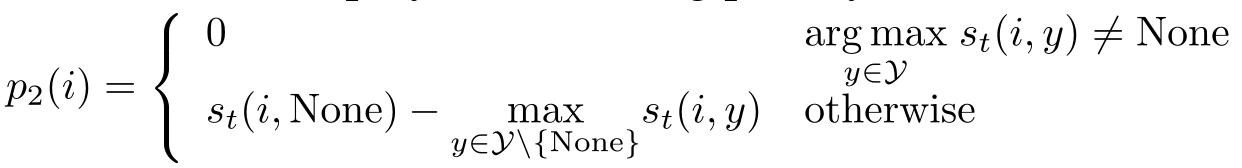
为了利用事件共指模型中的每个约束，我们定义了一个罚函数，如果两个跨度I和j违反了一个约束，则罚函数会对它们施加惩罚，其中I是回指，j是I的候选先行词。

对于P1，**惩罚函数P1**的计算如下:



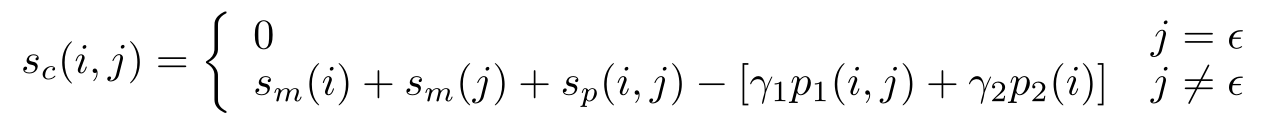
其中yi = argmaxytst(i，yt)和yj = arg maxytst(j，yt)。非正式地，p1返回确保I和j具有相同事件子类型所需的最小调整量。

对于P2，**惩罚函数P2**的计算如下:



其中Y是可能的子类型的集合。非正式地说，p2返回确保I的子类型不是NONE所需的最小调整量。

最后，通过在共指模型中使用这两个惩罚函数更新Sc(等式1):



**Exploiting Entity Coreference Information**

Computing Entity Coreference Chains

为了计算实体共指链，我们使用两个模型

**Independent model** 第一个模型是一个独立的模型，它被训练为独立于基线事件共指模型来计算给定文档中实体提及的共指链。

实体共指模型也有三层:

1. 跨度表示层，学习实体提及对应的跨度；
2. 实体提及预测层，其向每个跨度分配从预定义实体类型的储存库中获取的实体类型，或者如果跨度不对应于实体提及，则分配NONE
3. 实体共指层，其计算用其预测的实体类型扩充的跨度上的共指链接。

损失函数是两个损失的加权组合，一个对应于预测实体提及及其实体类型的实体提及检测，另一个对应于实体共指，其损失的定义方式与用于事件共指的相似。

**Joint model** 独立模型的一个**缺点**是实体共指和事件共指之间的交互很少。为了更紧密地耦合这两个任务，我们联合训练独立实体共指模型和基线事件共指模型。

当以联合方式训练时，独立和基线共享跨度表示层，联合使用的损失函数是独立和基线使用的损失函数的加权组合。

**Incorporating Entity Coreference Information**

**两种方法：**

**As a hard constraint** 使用实体共指信息来创建实体共指和事件共指之间的约束，该约束指定 如果两个事件提及的对应参数不是实体共指，则它们不能是共指。

**As features** 使用实体共指信息来计算三个二进制特征，用于训练事件共指模型。第一个特征编码考虑中的两个事件提及是否有任何共同的语义角色。第二个特征编码它们的任何具有相同语义角色的参数是否不是实体相关的。第三个特征是第二个特征的补充:当且仅当第二个特征的值为0时，它的值为1。

这两种方法可以与两种模型中的任何一种结合使用。作者评估了四种组合，即**Independent/Feature (IF),Independent/Constraint (IC), Joint/Feature (JF), 和 Joint/Constraint (JC).**

**评估**

**数据集：**

**（训练）LDC2015E29、E68、E73、E94和LDC2016E72**

**（测试）KBP 2017**

**评估指标**

**事件共指：AVG-F**

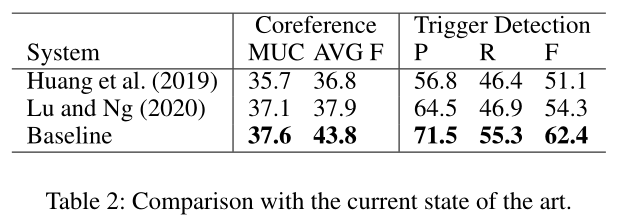
**触发器检测：召回率(R)、精确度(P)和F-score (F)**

**实体共指：CoNLL分数**

**实体检测：F-score (F)**

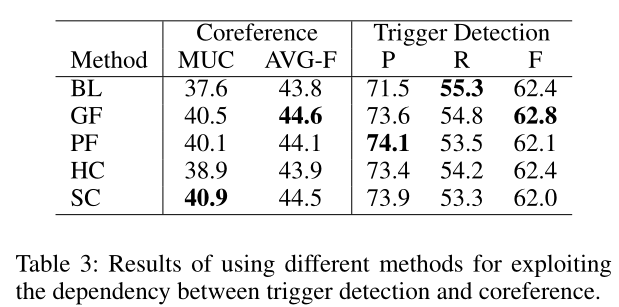
**结果**

**问题1：基于span的模型对于事件共指解析的效果如何？**



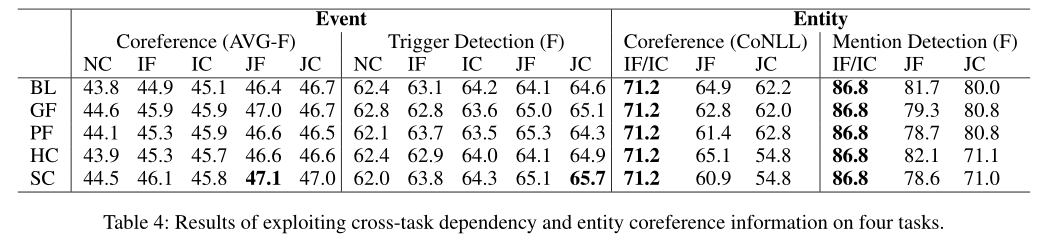
**问题2：在基于span的事件共指模型中，利用事件共指和触发器检测之间的依赖性的最佳方式是什么？**

作者通过四种方法应用于基线模型来了解利用事件共指和触发器检测之间的依赖性是否能够提高共指性能。



**问题3：实体共指信息有利于事件共指解决吗？**

**通过评估四种组合，来检验实体共指信息是否有利于事件共指。**



列2-5显示了利用实体共指信息的四种组合。为了进行比较，“NC”列中显示了没有实体共指信息的结果。

联合结果(JF和JC)优于独立结果(IF和IC)，而独立结果又优于不使用实体共指信息时的结果。