**Active Learning for Coreference Resolution using Discrete Annotation**

会议：ACL2020

代码：<https://github.com/belindal/discrete-active-learning-coref>

**摘要**

我们通过要求注释者识别提出的提及对是否被认为是不恰当的，改进了基于对注释的主动学习的共指解析。 与用于选择要标记哪些示例的新颖提及的聚类算法结合使用时，这种简单的修改在按注解预算获得的性能方面要高效得多。 在使用现有基准共参考数据集进行的实验中，我们表明，此附加问题的信号导致每条人工注释小时的性能显着提高。将来的工作可以使用我们的注释协议为新域有效开发共指模型。 我们的代码是公开可用的。

**1引言**

共指解析（两个词是否指向同一个实体）是将照应性表达解析为其先例的任务（请参见图1）。 它在下游应用中通常是必需的，例如问答（Dasigi等，2019）或机器翻译（Stanovsky等，2019）。 详尽地注释共指是一个昂贵的过程，因为它需要跨较长的文本段落跟踪共指链。 例如，在新闻报导中，重要实体在引入后可能会引用许多段落。

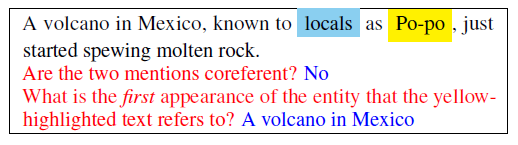


图1：离散注释。 注释器显示该文档，一个跨度（黄色）和该跨度的预测前例（蓝色）。 如果对共同指称问题的回答是否定的（即跨度不是共指），我们将提出一个后续问题（“实体的首次出现是什么？”），以提供其他具有成本效益的信号。 我们的注释界面可以在附录的图5中看到。

主动学习是一种旨在通过注释样本（这对学习过程最有益）来降低成本的技术，而不是完全标记大型固定训练集。 主动学习包括两个部分：（1）特定于任务的学习算法，和（2）迭代样本选择算法，该算法检查在先前迭代中训练的模型的性能，并选择要添加到注释训练集的样本。事实证明，该方法可成功完成资源不足领域的各种任务（Garrette和Baldridge，2013年; Kholghi等人，2015年; Syed等人，2016年，2017年）。

Sachan等（2015年）表明，主动学习可以用于共指解决任务。 他们使用GOLD数据来模拟成对的人类注释，其中两个实体的注释被注释为是否为coreferring（请参见图1中的第一个问题）。

在本文中，我们为主动学习提出了两项​​改进，以实现共指解析。 首先，我们引入离散注释的概念（第3节），它通过引入一个简单的附加问题来增强成对注释：如果用户认为这两个提及是非共指的，则要求他们标记其中一个提及的首次出现（ 参见图1中的第二个问题）。 我们表明，这种简单的加法有几个积极的含义。 对于批注者而言，反馈相对容易，并提供有意义的信号，从而大大减少了对文档进行完全标记所需的批注数量。

其次，我们介绍提及聚类（第4节）。 在选择要标记的下一个提及内容时，我们会考虑汇总模型预测属于同一类的所有先例。这避免了重复标记，而重复标记将像以前的方法那样单独验证同一集群中的每个提及对。

我们使用现有的GOLD数据作为用户标签，对几种样本选择算法进行了实验，结果表明，我们的两项工作都显着提高了CoNLL2012数据集的性能（Pradhan等，2012）。 总体而言，我们的主动学习方法为成对注释提供了一种更好的替代方法，以实现共指解析，从而在给定注释预算下实现了性能更好的模型。

**2背景**

我们的工作依赖于两个主要组成部分：共指解析模型和样本选择算法。

**共指解析模型** 我们使用Lee等人介绍的跨度排名模型。（2017），后来在AllenNLP框架中实施（Gardner等人，2018）。 该模型计算文档中所有可能跨度i的跨度嵌入，并使用它们来计算在所有候选前项集合上的概率分布，其中是一个虚拟前提，表示跨度i没有任何前提。 该模型不需要额外的资源，例如句法依赖性或命名实体识别，因此非常适合于资源匮乏的领域的主动学习方案。

样本选择算法先前用于共指分辨率标注的方法大多使用成对选择，其中一对标注被显示给人类注释者，以标记它们是否是共同参考（Gasperin，2009； Laws等，2012； Zhao和Ng ，2014； Sachan等人，2015）。 为了将这些二进制注释合并到它们的聚类共引用模型中，Sachan等人（2015）引入了必须链接和不能链接惩罚的概念，我们将在第4节中进行描述和扩展。

**3离散注释**

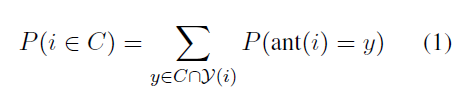
在离散注释中，如图1所示，我们为注释器提供了一个文档，其中文档的最小确定范围i（在示例中为“ Po-po”）和i的模型预测的前项A（i）（“ 当地人”），被高亮。与成对注释相似，首先会询问注释器i和A（​​i）是否互相关。如果他们的回答是肯定的，那么我们继续进行下一个示例。 否则，我们将偏离成对抽样，并要求注释者将i的先例（“墨西哥的火山”）标记为后续问题。如果我不是一个有效的提及者，或者如果它在文档中没有前身，那么注释者可以不回答后续问题。 有关更多示例注释，请参见附录中的图5。

在第5节中，我们展示了离散注释在几个方面都优于经典的成对注释。 首先，它可以更好地利用人类注释时间，因为注释者通常需要解决所提到的内容的前提才能回答第一个问题。 例如，识别“ Po-po”是指火山，而不是当地人。 其次，我们发现离散注释更适合于提及排名模型（Lee et al。，2017），该模型为每个提及分配最可能的前因，就像注释者在离散注释中所做的一样。

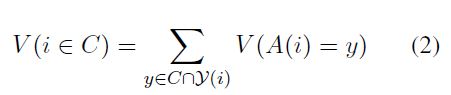
**4提法聚类**

我们通过将流行的主动学习选择器（如熵或按组查询）（Settles，2010）应用于跨度聚类，尝试了三种选择技术。 因为我们的模型输出先验概率和预测，所以我们希望汇总这些输出，以便每个提及簇只有一个概率，而不是每个先例都有一个。我们举一个例子来说明这一点：假设跨度i的最可能的两个先例y1和y2。 在方案1中，预测y1和y2聚集在一起，在方案2中，预测将它们分开聚集。跨度i在场景1中应该具有“较高的确定性”（因此不太可能被主动学习选择），因为它的两个最有可能的先行词都暗示着相同的聚类，而在场景2中，选择会导致一个不同的下游集群。因此，我们不是简单地使用原始概率i指代特定的先例，而是使用概率i属于某个聚类。 这意味着在场景1中“联合”建模y1和y2，在场景2中分别建模。

形式上，我们通过将P（ant（i）= y）相加来计算跨度i属于群集C的概率对于属于某个群集C的所有y，因为i在群集中具有先行条件也必然意味着i也在该群集中。 这使我们能够将预测的先例概率转换为集群内概率：



类似地，对于按委员会查询，我们汇总预测，以使每个集群有一票，而不是每个先例一票：



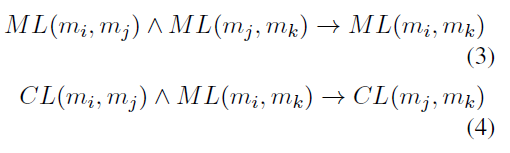
其中表示将y选为i的先例的模型的数量。

我们在等式1和2中使用的聚类信息是根据模型预测的标签和通过主动学习查询的标签的组合计算的。 以前没有出现在群集中的先例被视为单例群集。

此外，为了在选择过程中尊重用户注释，我们必须跟踪所有先例的注释。 为此，我们使用了由Sachan等人介绍的提及之间的必须链接（ML；如果判断为两个引用是coreferent）和不能链接（CL；如果判断为两个引用不是coreferent）的概念（2015），并根据我们的目的对其进行调整。 具体而言，在离散设置中，我们按以下方式构建链接：如果用户认为该对为coreferent，则将其添加到ML。 否则，它将被添加到CL，而用户更正的对（来自第二个问题）将始终添加到ML。

此外，我们使用这些链接来指导我们如何选择下一个要查询的内容。 例如，如果跨度m1和m2之间存在CL关系，则我们将不太可能查询m1，因为我们可以更确定m1的先行条件（not m2）。正式地，我们根据链接关系修改概率并投票，这会影响选择器不确定性评分。

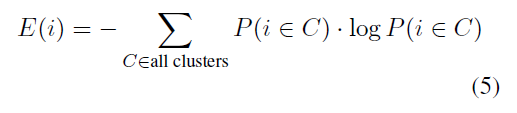
最后，继(Sachan等人，2015)之后，我们施加了转折性约束，这使得我们能够对超出注释过程中明确指出的链接进行建模。



但是，在每次主动学习迭代之后重新计算这些闭包可能会非常无效率。取而代之的是，每次添加新链接时，我们仅通过添加最少数量的必要链接来逐步维护封闭关系，从而逐步建立封闭关系。

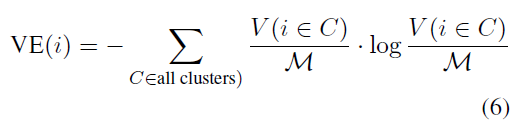
我们尝试以下聚类选择技术：

聚类熵 我们计算聚类概率的熵，并选择聚类熵最高的提及：



其中如公式1所述。

**Clustered query-by-committee** 我们训练M个模型（具有不同的随机种子），并选择具有最高聚类投票熵的提及：



使用等式2中定义的在整个群集中计数的选票。

**最少的核心簇集提及/最核心的非簇集提及（LCC / MCU）** 我们旨在选择模型对其预测最缺乏信心的跨度子集。 对于每个被分配了聚类Ci的跨度i，我们计算得分，并选择最小的n个跨度。对于每个单例j，我们给出一个“未聚类”的分数，并选择最大的m个跨度。 用公式1计算。

**5评估**

我们使用英语CoNLL-2012共参考数据集比较了离散注释和成对注释（Pradhan等，2012）。 按照Sachan等（2015），我们进行实验，从金标注中模拟用户的判断。

**注释时间估计**为了比较成对问题和离散问题之间的注释时间，我们从NLP中有背景的7个内部注释者那里收集了8个30分钟的会话。 注释者被要求在那30分钟内回答尽可能多的实例。 我们另外请1个注释者在30分钟内仅注释离散问题。 为了尽可能地具有代表性，这些实验的主动学习查询是从主动学习的各个阶段中取样的（请参见表1）。 平均而言，注释者在一个会话中完成了大约67个问题，其中一半回答为否，因此需要附加的离散问题。 总体而言，这些估计值取决于826个带注释的答案。我们的注释界面是公开可用的，请参见附录中图5中的示例。

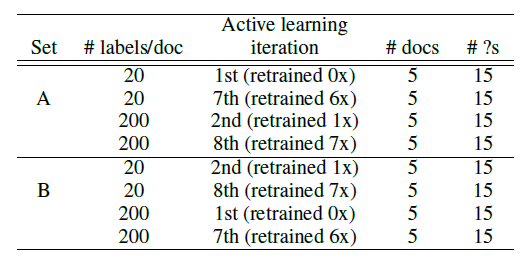


表1：定时实验采样。 对于这两个数据集，我们从20个文档中总共收集了60个主动学习问题。 我们针对4个类别分别收集了5个文档和15个问题：每个文档训练有很多/很少标签，并且在主动学习过程中处于早期/晚期。 从一个迭代中随机抽取了15个问题。

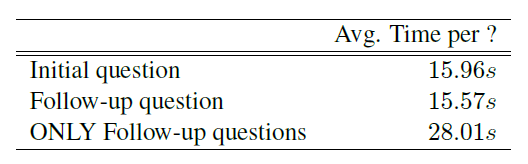


表2：初始成对问题，离散后续问题和离散问题的平均注释时间。

时间结果如表2所示。回答最初的成对问题之后的离散问题与回答第一个问题大约需要相同的时间（约16秒）。 此外，仅回答离散的问题每个问题花费了28:01s的时间，这证实了，如果回答肯定，最初的成对问题确实可以节省注释者的时间。

在以下实验中，当计算总注释时间时，我们将使用这些度量来校准成对和离散的后续问题。

**基线**我们使用熵选择器为成对注释实现基线。 我们还通过随机选择实现了两个离散的注释基线。 部分标记的基准遵循标准的主动学习训练循环，但会随机选择下一个要标记的标记。 完全标记的基线通过将注释时间t作为输入并随机选择用户可以在t小时内仅使用离散注释完全标记的一组文档来创建训练数据的子集。 通过将完全标记的基准与我们的主动学习结果进行比较，我们可以确定主动学习相对于彻底标记文档是否有效。

**超参数**我们使用Lee等人的AllenNLP实现中的模型超参数。 （2017）。 在添加标签之前，我们最多要耐心训练2个20epochs。 添加所有文档后，我们将从头开始进行培训。由于内存的限制，我们使用M = 3个模型的委员会查询。 对于LCC / MCU，给定每个文档L个注释，我们将注释在群集和单例之间平均分配。

**结果**图2绘制了第4节中使用各种选择器的离散注释的性能，以及根据我们的时序实验校准的成对注释的性能。在所有图中，我们将MUC，B3和CEAFe报告为平均F1分数。

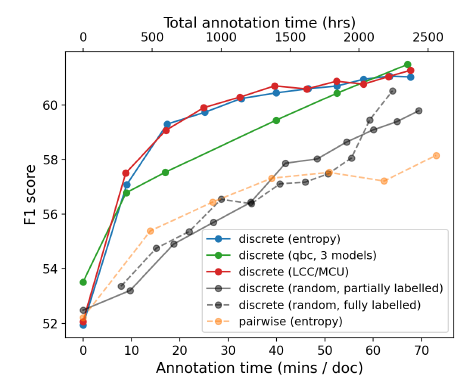


图2：比较各种选择器以进行离散注释和成对注释（橙色虚线）。

这三个非随机主动学习框架的表现优于完全标注的基线,表明注释预算有限时，主动学习对于共指解析更为有效。

最值得注意的是，图2显示了每种非随机离散选择协议都优于成对注释。 在性能差距最大的地方（每个文档大于15分钟），我们的绝对F1始终比成对选择提高4％。

**6分析**

离散注释优于成对基线的主要原因是完全标记文档所需的成对注释数量远大于离散注释的数量。 在每个提及有201个候选人的平均开发文档中，完全标记文档所需的成对查询数为15050，而离散查询的最大数量仅为201（即要求每个提及的前提）。 因此，普通文档可以通过离散注释完全注释的时间是使用成对注释完全标记它的时间的2.6％，这表明我们的框架也是可行的穷举注释方案。

进一步的分析表明，离散选择的改进部分归因于更好地利用注释时间来提高提及检测的准确性（图3）和代词的解析度（图4），在这种情况下，我们仅对带有代词的词组进行了性能测量， spaCy tagger（Honnibal和Montani，2017年）。

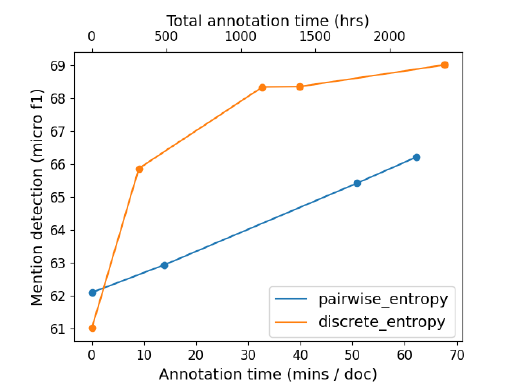


图3：每个人注释时间的成对选择和离散选择的提及检测精度（在document-micro F1中）。

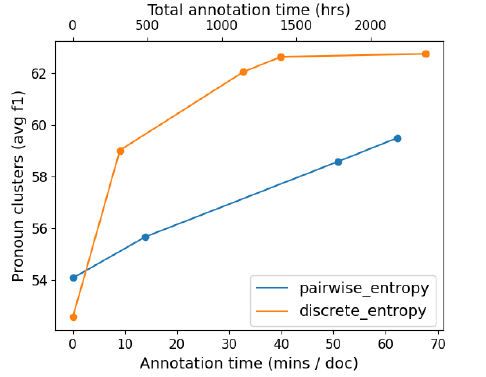


图4：每个人类注释时间的成对与离散选择的代词解析准确度（平均F1）。

最后，表3显示了我们离散注释框架的消融，显示了范例的每个组件的贡献。

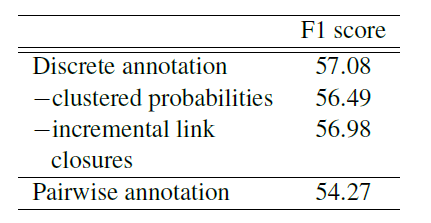


表3：在单个点（315个注解小时）上不同模型元素的消融。 熵选择器用于所有实验。

**7讨论和结论**

我们提出了离散标注，这是在资源少的领域主动学习共指分辨时成对标注的一种有吸引力的替代方法。通过在注解界面中添加一个简单的问题，我们每个人注解小时获得了明显更好的模型。 此外，我们引入了一种聚类技术，可在注释过程中进一步优化样本选择。 更广泛地说，我们的工作表明注释接口的改进可以引起响应，相对于所投入的注释时间而言，响应相对于获得的性能更有效。